Machine learning classification of entrepreneurs in British historical census data

Piero Montebruno[a], Robert J. Bennett[a,*], Harry Smith[a], Carry van Lieshout[a]

[a] *University of Cambridge, Department of Geography and Cambridge Group for the History of Population and Social Structure, Downing Place, Cambridge, CB2 3EN, UK.*

Abstract

This paper presents a binary classification of entrepreneurs in British historical data based on the recent availability of big data from the I-CeM dataset. The main task of the paper is to attribute an employment status to individuals that did not fully report entrepreneur status in earlier censuses (1851-1881). The paper assesses the accuracy of different classifiers and machine learning algorithms, including Deep Learning, for this classification problem. We first adopt a ground-truth dataset from the later censuses to train the computer with a Logistic Regression (which is standard in the literature for this kind of binary classification) to recognize entrepreneurs distinct from non-entrepreneurs (i.e. workers). Our initial accuracy for this base-line method is 0.74. We compare the Logistic Regression with ten optimized machine learning algorithms: Nearest Neighbors, Linear and Radial Support Vector Machine, Gaussian Process, Decision Tree, Random Forest, Neural Network, AdaBoost, Naive Bayes, and Quadratic Discriminant Analysis. The best results are boosting and ensemble methods. AdaBoost achieves an accuracy of 0.95. Deep-Learning, as a standalone category of algorithms, further improves accuracy to 0.96 without using the rich text-data that characterizes the OccString feature, a string of up to 500 characters with the full occupational statement of each individual collected in the earlier censuses. Finally, and now using this OccString feature, we implement both shallow (bag-of-words algorithm) learning and Deep Learning (Recurrent Neural Network with a Long Short-Term Memory layer) algorithms. These

[⋆]

Declarations of interest: none

[*] Corresponding author: Tel.: +44-01223-339957; fax: +44-01223-333392;
Email addresses: pfm27@cam.ac.uk (Piero Montebruno), rjb7@cam.ac.uk (Robert J. Bennett), hjs57@cam.ac.uk (Harry Smith), cv313@cam.ac.uk (Carry van Lieshout)

methods all achieve accuracies above 0.99 with Deep Learning Recurrent Neural Network as the best model with an accuracy of 0.9978. The results show that standard algorithms for classification can be outperformed by machine learning algorithms. This confirms the value of extending the techniques traditionally used in the literature for this type of classification problem.

Keywords: machine learning, deep learning, logistic regression, classification, big data, census

1. Introduction

Modern information processing techniques are as applicable to classifying and identifying patterns in historical data as they are to modern data. An important historical question has been 'who were entrepreneurs in the past?' This is a first and essential step towards identifying their characteristics and understanding their behaviour. The analysis of historical developments in entrepreneurship has lacked until recently sufficient data to be confident about the scale of historical activity and trends over time. After major efforts of transcription and data coding large scale, historical sources are now becoming available that allow entrepreneurs to be identified in the past from their descriptions of themselves. In England and Wales, a digitized version of the Victorian censuses over 1851-1911 has become available through the I-CeM database (Higgs and Schürer, 2014; Schürer et al., 2015). This has been enhanced in a supplementary database (the British Business Census of Entrepreneurs, BBCE) that extracts the members of the population who can be identified as entrepreneurs (Bennett et al., 2019). This provides a new resource for information analysis and also introduces scope to make long-term comparisons between modern and previous historical patterns. Unfortunately for the first four of these censuses (1851-81), accounting for nearly 80 million people, only a limited question referring to employers was used by the census administrators which does not allow direct and full identification of all entrepreneurs.

This paper studies the methodological challenge of classifying the 'employment status' of individuals as entrepreneurs (Ents) or workers (Ws) from the information that was self-reported in the census. Classification methods are assessed that are based on the individuals' demographics and also the descriptive text of their occupational activities in the archival records of the Census Enumerators Books (CEBs). Many learning methods have been developed in information science for related classifications; e.g. binary linkage (Boutell et al., 2004), classifier chains (Read et al., 2011), label powerset (Tsoumakas et al., 2011), rankings by pairwise comparison (Hüllermeier et al., 2008; Fürnkranz et al., 2008). These developments have expanded the focus in textual processing from title searches and tagging (Hu et al., 2006) to multiple tag interactions (Murthy and Gross, 2017; Al-Salemi et al., 2019; Tang et al., 2019), complex text interlinkages for result caching (Kucukyilmaz et al., 2017), deep textual semantic interactions (Kastrati et al., 2019), and attempts to identify sentiments through textual recurrence (Abdi et al., 2019). There is a rapidly growing literature on machine learning in the information sciences. However, there have been few applications to economic history. Schürer et al. (2015) developed a computerized classification method for the same 1851-1911 I-CeM data as used in this paper: to standardize and code occupational titles and also birthplace descriptors. This uses dictionaries of occupations and birthplaces and then develops a hierarchical system of matching to link actual terminology to dictionary terms. However, this is an artificial intelligence (AI) and not a machine learning (ML) method, though it tackles a similar problem to that here. Other applications of ML to related social science questions have used standard information science techniques such as Bayes Networks (Tang et al., 2016), used by Alvarez-Galvez (2016), to tackle interrelationships between socioeconomic status and health in Europe, or other machine learning methods used by Su and Meng (2016) to perform automated text analysis of online forums to assess the response to China's government policies, generalized boosting used by Reichenberg and Berglund (2019) to overcome some of the deficiencies of an inverse-probability weighting analysis, and structured learning used by Katz and Levin (2018) to classify individuals into types of political supporters using ML, based on joint responses to eight questions while estimating the association between each item and support dimension.

Our methodology first uses numerical and categorical variables from individuals' demographics given in the census and then applies text-based methods using the unique occupational string descriptions available in the earlier censuses. This classification of the population is of importance for understanding the scale and trends of entrepreneurship. The economic theory of entrepreneurship relies on the classification of individuals as Employer (E) and Own-account (OA) which was the Victorian census term for those proprietors operating on their own with no employees, as distinct from Ws. The sum of E and OA gives all self-employed, which following Parker (2004) and Blanchflower and Oswald (1998) we use as the definition of all Ents. The methodology developed here for entrepreneurs is focused on evaluating alternative estimation methods for this classification. However, the paper has broader relevance for any classification process attempted in other disciplines. Thus, it is not restricted to historical data.

The paper uses new developments in AI. AI refers to computers' thinking as humans do; as defined by The Editors of the American Heritage Dictionaries (2011), the verb to think can be defined as: "To exercise the power of reason, as by conceiving ideas, drawing inferences, and using judgment". This can be expanded by using training that involves known patterns as the input of the learning process where the patterns do not have explicit rules when computerized this is called a form of machine learning (ML). The process can be further expanded to deep learning (DL) when the model used is a distillation over several layers, or filters, where each attempts a better representation of the data. This is often called a neural network because its inspiration comes from understanding how the brain learns, though neural networks are not themselves considered a representation of the brain. François Chollet (2018) provides the following useful relation:

$$\text{Artificial Intelligence} \supset \text{Machine Learning} \supset \text{Deep Learning}$$

The methodology developed in this paper uses AI, ML, and DL. The historical mid-Victorian censuses in the UK that are now available as a digital database were collected in two formats: first, for 1851-81 a question was used that sought to distinguish employers and 'masters'; second, for 1891-1911 the question was modified to ask individuals explicitly to identify themselves as employers, own account, or workers: termed their 'employment status'. Hence, for the later period, the question attempted to collect full information on all entrepreneurs (as employers or own account) for the whole population. The change in the questions was a response to pressures from social scientists led by Charles Booth and Alfred Marshall, that the census administrators (General Register Office: GRO) should introduce a new question that identified the self-employed (Treasury Committee, 1890; see also Higgs, 2004).

The result of this change was a major improvement in census design as it provided a separate classifier that explicitly identified entrepreneurs, which was additional to their textual description of their occupation but also a fundamental loss of information as the 500-character occupational string was reduced to a 70 character one (see below for a detailed discussion). As a result, there is a discontinuity between the earlier censuses before 1891 where the census question provides potentially full coverage of employers from the string "employing..." in the OccString feature, but only partial coverage of own account for those cases where they identified themselves as the string 'masters' in the same OccString. The term 'master' had a historical meaning for those trained or apprenticed in some trades who could operate alone or employ others, but it was a term that was obsolete in many occupations by the mid-Victorian period, whilst in other occupations 'master' had never been used (for example in professions, commerce, transport, and many retail trades). Indeed, in 1851 only about 6 per cent of entrepreneurs used the term master, which fell to about 3-4 per cent by 1881 (Bennett et al., 2018).

The classification problem that we tackle is: can the explicit information in the later censuses on ‘employment status’ be used to train a classifier to identify entrepreneurs in the early censuses using their demographics or/and their textual responses to the question on employers, masters, and other occupations? Also, can information gathered from a subsample of the early censuses be used to train a classifier to generalize and identify entrepreneurs using standard demographic or/and text features? Finding a way to estimate entrepreneurial status for this early period is an important challenge since the later census questions align closely with modern censuses, thus allowing a continuous series of to be developed from 1891 to the present. Having an available benchmark for entrepreneurial status for 1851-81 allows the time series to be extended from 1851 up to the present, and it would also help develop long-term comparisons backwards to earlier periods before 1851.

Despite the progressive adoption of machine learning, this paper is one of the first to apply machine learning in a historical setting. Moreover, this use of machine learning solves a methodological gap in the classification of millions of individuals that on the night of each census responded with valuable demographic and economic information. In this paper we describe the classification problem, present the methodology for applying ML, and test the performance of different ML algorithms.

2. Research objective

The machine learning method we develop seeks to tackle a binary classification problem (if the labels are W and Ent), or a multi-class classification problem (if the labels are W, E and OA) (Boutell et al., 2004; Tsoumakas et al., 2011; Read et al., 2011). We test the performance of different ML algorithms against a traditional probability-based model using logistic regression (LR). LR is a standard in the literature for any binary classification(Cameron and Trivedi, 2005) and has been used in many information processing applications. It has also been the algorithm of election previously used to tackle the problem at hand by (Bennett et al., 2018, 2019). It is used here as a benchmark for comparison.

LR estimates the probability or likelihood of being in one of two alternatives. In our case, a binary classification of entrepreneurs should estimate the probability of being an entrepreneur in our data. As Chollet (2018) poses, LR is the "hello world" of modern machine learning. Our research objective is to apply ML algorithms not traditionally used and compare the performance of these algorithms for classification of entrepreneurs for British historical census data. We implement a range of new methods in the field of information science to test and compare ML algorithms that can potentially outperform the standard LR. Thus, our research question is: can alternative methods in ML exceed in performance the LR in our binary classification of entrepreneurs; and which alternative methods give the best performance. In an appendix, we also explore the multi-class classification of individuals (as Worker, Employer, and Own account).

The area of ML that we develop can be understood as *predictive* or *supervized* process of learning a mapping from inputs $\mathbf{x}$ to outputs y given a labeled set of inputs pairs $D = \{(\mathbf{x_n}, y_n)\}_{n=1}^{N}$ whith D the *training set*, and N the number of training examples (Murphy, 2012). In ML the inputs $\mathbf{x}$ are *features* (or *attributes*) while the outputs y are *labels* (or *targets*). When y is *nominal* or *unordered-categorical* with j categories and j goes from 1 to C, the problem is *classification* or *pattern recognition* (Murphy, 2012). If $C = 2$, the classification is *binary* and y is taken to be $\{0, 1\}$. If $C > 2$, the classification is *multi-class* (Murphy, 2012) . Traditional ML follows a method called *function approximation* where it is assumed that $y = f(x)$ for some unknown function f and the learning process is aimed at estimating the function f given a labeled training set, and then to make the predictions as follows:

$$\hat{y} = \hat{f}(\mathbf{x})$$

The process of calculating out-of-training-set predictions is then called *generalization.* Additionally, an algorithm that puts into action classification is called a *classifier.* It can also refer to the mathematical function performed by a classification algorithm, that maps features to labels. In our case, the chosen base-line classifier is the logit model (LR) for the binary responses and the MNL for the multi-class responses. This is a traditional classifier approach (e.g. Cheng and Hüllermeier

(2009)). The logit models are applied by using the attributes of each individual that most closely correlate with the entrepreneurial status where this was recorded in the later censuses. This logit estimate is then applied to the earlier censuses to classify individuals where only partial records of entrepreneur status were recorded. This gives the binary probability of being an entrepreneur (Ent) or not (W).

3. Methodology

In our previous work, we have applied two models, see for instance, Bennett et al. (2018) and Bennett et al. (2019). The two theoretical and original models are:

Model	Classifier	Type	Labels
Logistic Regression	LR	Binary	W, Ent
Multinomial Regression	MNL	Multi-class	W, E, OA

We have used LR and MNL to classify entrepreneurial (binary) and employment (multi-class). Both models have been useful, feasible and rewarding as a very standard way of ML. We have used them in data- and time-intensive settings with approximately 120,000,000 data points. In particular we took labeled data to estimate the coefficients and then predict or generalize to unlabeled data. We use an LR or an MNL classifier because they are, as stated above, the most used in the literature. But ML is increasing dramatically the performance of new algorithms and classifiers to solve analogous problems, thus we want to test in a controlled environment the performance of the LR (we do not use MNL for simplicity and clarity in the main figures of the paper as it is easier to visualize and present the results for the LR, but we present the main MNL results in tables and assess the differing mathematical characteristics in an appendix). We use golden data sets and the simplest metric possible (accuracy) to test different models. We present the comparison of accuracy up front for ease of understanding, but the main discussions and conclusions are to be found later in the paper. The first set of models are as follow:

Model	Classifier	Type	Labels	Features	Dataset	Tr/T set	Accur
1891 LR	Logistic Regression	Binary	W, Ent (responses recorded in the census)	SubOccode, RSD Density, Age, Sex, Marital status, Relationship to the head, Servants (not OccString)	1,000 W, 1,000 Ent from 1891	Tr: random 60%, T: random 40%	0.74
1851 LR	Logistic Regression	Binary	W, Ent (clerical responses using strings)	the same	1,000 W, 1,000 Ent from 1851	the same	0.82
1851 AdaBoost	AdaBoost	Binary	the same	the same	maximum possible set of extracted. 70,872 W, 70,872 Ent from 1851	the same	0.95
1851 Bag of words	AdaBoost	Binary	the same	the same + OccString (Bag of words)	the same	the same	0.9949

The 1891 LR uses a dataset where labels were self-reported by the individuals as a result of the question that included now a three-column self-reporting employment status with the labels: W, E, OA. We use the simpler: W, Ent, where Ent is E or OA. This model attempts to test the performance of the LR in this ground-truth dataset. We then switch to an 1851 dataset where the labels have been attributed by a clerical human-led intervention that used all available information including census names and non-census genealogy, internet or third-party information like Directors Directories, Chamber of Commerce data, and other historical sources. The 1851 LR is the base model where the labels are clerical, and the classifier is the standard LR. This is applied to a sample dataset of 1,000 individuals. 1851 AdaBoost is the same model but with two innovations: first, as classifier AdaBoost which is the best performing in the two features graphical ten-classifier comparison below, and second we have expanded the dataset to the maximum possible set of “extracted”—the ones where the clerical attribution of labels was guided by the employer and master titles in the BBCE which implies 70,872 W and 70,872 Ent. The 1851 Bag of words is the same model but with the addition of using the OccString features, only available for the censuses 1851, 1861, 1871, 1881, but not for the later censuses where the Booth & Marshall three-column self-reporting classification reduced the strings from up to 500 characters to less than 70 producing a fundamental loss of information. The 1851 Bag of words model outperforms any other model, inclusive of Deep Learning without OccString (as it will become clearer below Deep Learning with OccString performs even better at the top of all our tested methods), showing the importance of the feature OccString and its ML use, which suggests a major focus for the future avenues of research on the issue. The bag of word method relies on splitting each instance (a sentence) into its component tokens (tokens meaning any word, number or punctuation sign) and acquiring some information about them like the count that each token repeats itself, both in an unstructured and unordered way, thus the term bag of words. The method starts from transforming each instance of OccString to a vector of length equal to the vocabulary of our corpus of OccStrings. The simplest way to do this, and the way that we choose, is to a natural language processor in scikit-learn called CountVectorize which transforms each instance of OccString into vector with the counts that each token repeats itself. For example, suppose the corpus of OccString answers were the following three instances: “Piano forte maker master”, “Boot and shoe maker employing 4 men”, “Iron mine proprietor employing 70 men & engineer empg [sic] 100 men total 170 men”. The vocabulary would be of 18 tokens that sorted would look like ‘100’:0,‘170’:1,...,‘and’:3,‘boot’:4,....,‘piano:14’,‘proprietor:15’,‘shoe’:16,‘total’:17. So each instance would be an 18-dimensional vector with the count of each vocabulary in the instance and zeros for not mentioned words. For example, the first instance would be [0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0] were the last one is in the position 14 which represents precisely the token ‘piano’. We use this CountVectorize tool from sciki-learn, and the maximum expanded dataset vocabulary is rather small with 5,706 tokens in the vocabulary so that there is no need of other techniques, for example, TfidfVectorizer or HashingVectorizer.

Next, the following are the settings of a comparison between the base-line LR model in a two-feature setting with iteratively SubOccode, Age, and RSD Density, with ten other ML classifiers:

Classifier	Type	Labels	Features	Dataset	Training set	Acc
Nearest Neighbors	Binary	W,Ent	SubOccode, Age	1,000 W, 1,000 Ent from 1851 with clerical responses using strings (not OccString)	training (random 60%), test (random 40%)	0.92
Nearest Neighbors	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.90
Nearest Neighbors	Binary	W,Ent	Age, RSD Density	the same	the same	0.66
Linear SVM	Binary	W,Ent	SubOccode, Age	the same	the same	0.78
Linear SVM	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.77
Linear SVM	Binary	W,Ent	Age, RSD Density	the same	the same	0.70
RBF SVM	Binary	W,Ent	SubOccode, Age	the same	the same	0.87
RBF SVM	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.89
RBF SVM	Binary	W,Ent	Age, RSD Density	the same	the same	0.72
Gaussian Process	Binary	W,Ent	SubOccode, Age	the same	the same	0.93
Gaussian Process	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.89
Gaussian Process	Binary	W,Ent	Age, RSD Density	the same	the same	0.73
Decision Tree	Binary	W,Ent	SubOccode, Age	the same	the same	0.93
Decision Tree	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.91
Decision Tree	Binary	W,Ent	Age, RSD Density	the same	the same	0.72
Random Forest	Binary	W,Ent	SubOccode, Age	the same	the same	0.93
Random Forest	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.91
Random Forest	Binary	W,Ent	Age, RSD Density	the same	the same	0.73
Neural Net	Binary	W,Ent	SubOccode, Age	the same	the same	0.84
Neural Net	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.78
Neural Net	Binary	W,Ent	Age, RSD Density	the same	the same	0.72
AdaBoost	Binary	W,Ent	SubOccode, Age	the same	the same	0.94
AdaBoost	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.90
AdaBoost	Binary	W,Ent	Age, RSD Density	the same	the same	0.72
Naive Bayes	Binary	W,Ent	SubOccode, Age	the same	the same	0.81
Naive Bayes	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.71
Naive Bayes	Binary	W,Ent	Age, RSD Density	the same	the same	0.71
QDA	Binary	W,Ent	SubOccode, Age	the same	the same	0.79
QDA	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.71
QDA	Binary	W,Ent	Age, RSD Density	the same	the same	0.71
LR (Baseline)	Binary	W,Ent	SubOccode, Age	the same	the same	0.76
LR (Baseline)	Binary	W,Ent	SubOccode, RSD Density	the same	the same	0.69
LR (Baseline)	Binary	W,Ent	Age, RSD Density	the same	the same	0.70

As said, each model is tested with three pairs of two variables from SubOccode, Age and RSD Density. SubOccode is a 844 classification derived from the occupation string provided by I-CeM database (Higgs and Schürer, 2014; Schürer et al., 2015) and can be understood as the minimal observation available in the OccString. It is not the OccString but its minimal information. For instance, for "Piano forte maker master" has SubOccode "Piano organ maker", "Boot and shoe maker employing 4 men" has "Shoe and boot makers (and repairers)", and "Iron mine proprietor employing 70 men & engineer empg [sic] 100 men total 170 men" has "Iron miner, quarrier". So the more information the string has, the greater the loss of information. Age is the age of the individual, and RSD (Registration Sub District) Density is the population density at the Registration Sub District geographical level. The ten models are compared with the LR (Baseline) at the bottom of the table. The best performing model is AdaBoost, that is why it is used in a full setting in 1851 AdaBoost model above.

The next box shows a completely different approach, Deep Learning. In Deep Learning we aim at two things: first to outperform 1851 AdaBoost model with the full set of features but not OccString, and then to outperform 1851 Bag of words with just OccString. The following model attempts and succeeds in the first task:

Model: Deep Learning; Architecture: Sequential Neural Network with 2 Dense Layers (16 hidden units) plus a rectified linear unit ("relu") activation and input shape of (906,) + 1 Dense Layer (1 hidden unit) plus a sigmoid activation; Optimizer: Root Mean Square Propagation (RMSprop); Loss: binary crossentropy; Metrics: Accuracy; Type: Binary; Labels: W, Ent; Features: SubOccode, RSD Density, Age, Sex, Marital status, Relationship to the head, Servants (not OccString); Dataset: maximum possible set of extracted. 70,872 W, 70,872 Ent; Epochs: 20; Batch size: 512; Accuracy: 0.96

In this case we use an architecture with just three layers, a RMSprop optimizer, a binary crossentropy loss function, 20 epochs and 512 batch size. This is the best performing model which does not use the OccString feature which is remarkable because it performs better than all the non-Deep Learning ML cases.

Next using only OccString, we build two additional DL models that do not use Bag of words but embed sentences in sequences of vectors: first a simple Dense architecture after flattening as follows:

Model: Deep Learning; Architecture: Sequential Neural Network with 1 Embedding Layer + 1 Flatten Layer + 1 Dense Layer (32 hidden units) plus a rectified linear unit ("relu") activation + 1 Dense Layer (1 hidden unit) plus a sigmoid activation; Optimizer: Root Mean Square Propagation (RMSprop); Loss: binary crossentropy; Metrics: Accuracy; Type: Binary; Labels: W, Ent; Features: OccString; Dataset: maximum possible set of extracted. 70,872 W, 70,872 Ent; Epochs: 10; Batch size: 32; Accuracy: 0.9964

And then a more tailored one, with a Recurrent Neural Network (RNN) with a Long short-term memory (LSTM) layer:

Model: Deep Learning; Architecture: Recurrent Neural Network (RNN) with 1 Embedding Layer + 1 Flatten Layer + 1 LSTM Layer + 1 Dense Layer (1 hidden unit) plus a sigmoid activation; Optimizer: Root Mean Square Propagation (RMSprop); Loss: binary crossentropy; Metrics: Accuracy; Type: Binary; Labels: W, Ent; Features: OccString; Dataset: maximum possible set of extracted. 70,872 W, 70,872 Ent; Epochs: 10; Batch size: 32; Accuracy: 0.9978

When working with text data, RNN (like SLTM or Gated Recurrent Unit Network, GRU) and convnets (Convolutional Networks) are to be preferred (Chollet, 2018). "The embedding layer is as a dictionary that maps integer indices (which stand for specific words) to dense vectors" in a relatively low-dimensional and learned from data manner (Chollet, 2018). This is strikingly different from the previous 1851 Bag of word model, where the encoding is "sparse, high-dimensional, and hardcoded" (Chollet, 2018). Hence, we have shown that LSTM with embedding of the occupational string but no other pre-processing as a bag of words, performs better than DL Dense and shallow learning 1851 Bag of words. This is one of the key results of this paper.

4. Data

The data that we seek to classify are the transcriptions of the 1851-1911 censuses of England and Wales as provided in Higgs and Schürer (2014) supplemented as in Bennett et al. (2019). For our purpose, we use only the non-farm population of entrepreneurs because the census data collected on farmers is sufficient to allow their entrepreneurial status to be identified without the need for machine learning or supplementation of census responses; hence they do not need the estimation processes discussed here (for an assessment of shifts in agrarian entrepreneurs see Montebruno et al. (2019a)). For non-farm entrepreneurs, Table 1 confirms that, since the means and the medians for each feature are statistically significantly different for each entrepreneur label class for the later censuses (in this case for 1891 as an example), a binary classification can be used based on the features of 1891 as a training set. All the t- and z-statistics of the two-sample t-test with equal variances and the two-sample Wilcoxon rank-sum, or Mann-Whitney, tests show that the difference between the means and the medians for the group are statistically significantly different from zero with p-values roughly equal to zero. At the same time, Table 2 shows a similar picture for classifying multiple attributes using the MNL with same features but now with labels 1 = Worker (W), 2 = Employer (E), and 3 = Own account (OA). Again the t- and z-statistics are all statistically significantly different from zero with corresponding p-values (almost) equal to zero. Hence, in our data, the attributes of Ents as a whole are statistically distinct from W, and E and OA are statistically distinct from each other and W.

5. Empirical analysis

The empirical analysis relies on three ground-truth (gold standard) datasets: "1891 1000 Ent", "1851 1000 Ent", and "1851 MAX(Extracted)" available for download in Mendeley Data (Montebruno et al., 2020). All the results of this paper can be replicated using those datasets. Our approach to the problem of classification of the 1851-81 censuses is to train the data with the known labels in the later 1891-1911 censuses, using the entrepreneur status that is fully reported in these later censuses (but not the earlier ones). Following the definition of predictive or supervized ML, we first develop a base-line model by approximating the unknown classification function with a logit (LR) model, using as training set the 1891 census where the labels 0 = Worker and 1 = Entrepreneur come from the reported employment status responses given in this census that are not available in earlier censuses. The LR classifier uses as classification features the following: the coding of the individual's occupational statement (SubOccode: See Bennett et al. (2018), for the list of the 844 occupational categories, which are sub-divisions of the I-CeM Occodes), Registration- sub-district (RSD) population Density (to use information of each individual's location), the individual's Age, Sex, Marital status, Relationship to the head of the household, and Number of servants in the household (which is a family resource surrogate). The method tests the accuracy of the LR for classifying individuals for whom their entrepreneur status is known from their 1891 census responses. The dataset has 1,000 entrepreneurs and 1,000 non-entrepreneurs (the training set is a random subset of 60% and the test set a random subset of the other 40% of the full datasets). The LR achieves an accuracy of 0.74 and a confusion matrix (see below for a full explanation) given in Figure 1. The full trained model is presented in Table 3 where a logit classifier is used and each weight ($\boldsymbol{w}$) and its $\boldsymbol{t}$-statistic are given. The same model but stripped of the SubOccode feature for computing efficiency is given in Table 4 where not only weights but also partial derivative marginal effects are provided. Note that the marginal effects are given only for the level variables and not for the squared, interaction and constant terms.

A similar procedure is followed with an MNL model with labels W/E/OA dropping the SubOccode feature for ease of computation with a dataset of 1,000 in each category (and similarly defined training set as above). Table 5 shows each weight ($\boldsymbol{w}$) and its $\boldsymbol{t}$-statistic, while Table 6 shows the partial derivative marginal effects for each variable in levels.

The performance of the method is improved by keeping the standard LR classifier but using a dataset for training and testing purposes from the 1851-81 censuses with labels derived not from full information of entrepreneurial status as in the later censuses where information of employment status was given in a second question but, instead, from clerical labeling by researchers of employment status using the occupation text strings of a large subsample of individuals as W and Ent. The occupational strings are contained in the Higgs and Schürer (2014) dataset, with the clerical coding of entrepreneur status derived from Bennett et al. (2019). The strings have terms such as House servant, "Piano forte maker master", and other examples introduced earlier. Here the dataset has 1,000 W, and 1,000 Ent from 1851. The use of this 1851 dataset to train the LR method increases performance to 0.82 for the accuracy of estimating entrepreneurial status, as shown in the confusion matrix Figure 2. The quest to keep improving on these methods is the main aim of the rest of the paper. As (Wolpert, 1996) established there is no universally best model (i.e. the no free lunch theorem), and the assumptions that work well in one problem do not necessarily work well in another. Thus, our aim is to actively look for better performance among the inherent uncertainties of ML algorithms applied to this new research problem.

6. Results

6.1. Classifier comparison

Using Python library scikit-learn (Pedregosa et al., 2011), we compare a range of alternative classifiers to the standard logistic regression based on coding by Varoquaux and Müller (2018) (under a 3-clause BSD License). Figure 6 shows the accuracy and 2-D predicted probability grid for the label being an entrepreneur (Ent) with 2,000 balanced random data points for ten classification algorithms: Nearest Neighbors, Linear Support Vector Machine (SVM), Radial Basis Function (RBF) SVM, Gaussian Process, Decision Tree, Random Forest, Neural Network (Net), Adaptive Boosting (AdaBoost), Naive Bayes, and Quadratic Discriminant Analysis (QDA) (see Zhang and Zhou (2007); Schapire and Singer (1999); Tang et al. (2016); Tong and Chang (2001); Wu et al. (2014); Alvarez-Galvez (2016); Freund and Schapire (1996); Friedman (2001); Murphy (2012)). In the first row of the figure the features are SubOccode and Age, in the second row SubOccode and RSD Density and SubOccode, and in the third Age and RSD Density. The purple circles are Ws and the green ones Ents. The color in the background is the 2-D predicted probability grid which means that when the grid is purple a test point will be classified as W and when the color is green a test point will be classified as Ent. The resulting probability patterns are strikingly similar to known patterns from the data (see Bennett et al. (2018)); e.g. that younger and older people are less entrepreneurial compared to middle years, and that lower density locations and certain SubOccodes are more entrepreneurial. The results show that the best performing methods are AdaBoost (which achieves accuracy of 0.94, 0.90, and 0.72, respectively, for all three rows), Decision Tree and Random Forest (both respectively 0.93, 0.91, and 0.72), and Gaussian Process (0.93, 0.89, 0.73). The standard LR performs systematically worse than almost all of the other methods tested here. Also, it can be seen that the best predictions are made using the features SubOccode and Age in combination, while the poorest predictions are made from Age and RSD Density. Of course, these are just three features for ease of visualization, but our final model selection uses all available features in the dataset.

6.2. Confusion matrix

The assessment of the classification of Ents and Ws by Age and Suboccode using the benchmark LR can be visualized in Figure 5. **(Note to referees: this fig should appear legible at full-page size if color is retained)**. There are two areas separated by a linear hyperplane. The first has a green zone on top, and to the right where the probability indicates a likelihood of being an Ent with two sets of individuals: green circles or True Positives (TP), true Ents predicted as Ents, and light purple crosses or False Positives (FP), true Ws predicted as Ents. The second area has a purple zone at the bottom, and to the left where the probability indicates a likelihood of being a W with again two sets of individuals: light green crosses or False Negatives (FN), true Ents predicted as Ws, and purple circles or True Negatives (TNs), true Ws predicted as Ws. Once we have selected the classifier, we run the AdaBoost method of Freund and Schapire (1996) using a parametrization suggested in Dawe (2018) (under a 3-clause BSD License) for the dataset with the maximum possible set of "extracted" (those labeled according to their strings or type of employment occupation codes) Ents (after excluding farmers). The method is now applied to the full 1851 labeled subsample of 70,872 Ws, 35,436 Es, and 35,436 OAs. In the binary classification problem, the following table shows at the bottom and in bold the predicted labels $\boldsymbol{(\widehat{-}, \widehat{+})}$, on the left and in bold the actual labels $\boldsymbol{(-, +)}$, four cells with TN, FP, FN, and TP:

	$\boldsymbol{N_{\widehat{-}}}$	$\boldsymbol{N_{\widehat{+}}}$	**TOTALS**
$-$	TN	FP	$\boldsymbol{N_-}$
$+$	FN	TP	$\boldsymbol{N_+}$
	$\widehat{-}$	$\widehat{+}$	

The confusion matrix is similar with the only difference that the number in the cells are rates over the previous table numbers after summation by rows. In particular it is important the sum of the rows or the true number of negatives, upper row or $\boldsymbol{N_- = TN + FP}$, and the true number of positives, lower row or $\boldsymbol{N_+ = FN + TP}$:

$-$	Specificity	False Alarm	(Denominator $\boldsymbol{N}_{-}$)
$+$	Missed Detection	Sensitivity	(Denominator $\boldsymbol{N}_{+}$)
	$\widehat{-}$	$\widehat{+}$	

Upper left, the Specificity Rate, $\boldsymbol{TN/N_{-}}$; upper right the False Alarm, Type I errors, or False Positive Rate, $\boldsymbol{FP/N_{-}}$; lower right, the Sensitivity, Recall, True Positive or Hit Rate, $\boldsymbol{TP/N_{+}}$; and lower left, the Missed Detection, Type II errors, or False Negative Rate, $\boldsymbol{FN/N_{+}}$. Not shown are the rates summing the columns for the "called" number of positives, right column or $\boldsymbol{N}_{\widehat{+}}$, and the "called" number of negatives, left column or $\boldsymbol{N}_{\widehat{-}}$. For example, Figure 3 shows an accuracy of 95% with TP of 27,561, TN of 26,267, FP 2104, and FN of 766. The AdaBoost classifier results in a reduced number of False Alarms (FP) and almost no Missed Detections (FN) as expected from Schapire and Singer (1999), Murphy (2012), and Al-Salemi et al. (2019) with a Sensitivity Rate of 97.3%, a Specificity Rate of 92.6%, a False Alarm Rate of only 7.4%, and a Missed Detection Rate of 2.7%. The Precision Rate of 92.9%, not shown in the confusion matrix because the denominator is the sum of the right column or $\boldsymbol{N}_{\widehat{+}}$, the predicted number of positives.

6.3. ROC curves

A further consideration is the extent of true positives and false negatives. The receiver operating characteristic (ROC) curve (Fawcett, 2006) is a means to assess this. Figure 7 uses the ROC plot: the Sensitivity Rate or True Positive Rate against the False Alarm or False Positive Rate at different thresholds $\boldsymbol{\tau}$ or cut-offs of the probability of being an Ent. If $\boldsymbol{\tau = 0}$ we are at the top right corner of Figure 7 where everyone is classified as an Ent so the True Positive and False Positive Rates both equal one as the $\boldsymbol{TP, FP > 0}$ while $\boldsymbol{FN, TN = 0}$. An analogous case, when $\boldsymbol{\tau = 1}$ is at the bottom left corner where everybody is classified as a W and both rates are now zero as $\boldsymbol{TP, FP = 0}$ while $\boldsymbol{FN, TN > 0}$. Along the diagonal and for different $\boldsymbol{\tau s}$ the two rates are equal as long as the Ent/W assignment is random. We plot the ROC curve for the following classifiers: RandomTrees (RT), RandomForest (RF), GradientBoosting (GBT) as both stand-alone methods and combined with LR following coding by Head (2018) under a 3-clause BSD License. The best classifier is the one which achieves the top left corner. Again, the preferred comparison classifier is a *boosting* method—as discussed in Friedman (2001) and James et al. (2013). In fact Gradient

Boosting associated with the LR has the purple or outer-most curve giving the best ROC curve among all the classifiers, see Figure 7b. Similar to *Section 4.1*, the second place is achieved by the *ensemble* method RT that relies on the *wisdom of the crowd*, see Géron (2017).

6.4. Bag of words

A bag of words, Chollet (2018), is the result of a two-stage process. First, tokenization or the breaking of text into units called tokens, and second, a vectorization or the association of numeric vectors with the generated tokens. The term bag "refers to the fact that [one is] dealing with a set of tokens rather than a list or sequence: the tokens have no specific order."[1] The data are the same maximum possible set of "extracted" Ents, but now the feature OccStrings is added to the same maximum "extracted" set. This is similar to many web-search algorithms (Kucukyilmaz et al., 2017) and title extraction (Hu et al., 2006), without using the semantic links between textual items used by Kastrati et al. (2019). This produces the confusion matrix in Figure 4 with an accuracy of more than 99% still using the AdaBoost parametrization suggested by Dawe (2018). This result shows the power of using the full occupational descriptor text in the form of a bag of words to solve this ML task.

6.5. Deep Learning

6.5.1. General features

Deep Neural Networks (DNN) are an important advance in the art of ML (McCulloch and Pitts, 1943; Rosenblatt, 1958; Rumelhart et al., 1988) which is particularly valuable for complex textual classification (Abdi et al., 2019; Kastrati et al., 2019). As suggested by Chollet (2018), a good metaphor of DL is the uncrumpling of a complicated manyfold of data. For example, imagine two sheets of colored paper: one green for Ents and one purple for Ws. Put them one on top of the other and crumple them into a ball of paper. Now you cannot tell them apart. DL consists of chains of geometric operations (underlying *tensor operations*) to uncrumple this ball of paper in order to separate—that is to classify—the Ents green sheet from the Ws purple one; or "finding neat representations for complex, highly folded manifolds" (Chollet, 2018). A simple architecture of DL should include an input of features, layers of data transformation parametrized by weights,

[1] As a hash in Perl. See Figure 6-2. A hash as a barrel of data. (Schwartz et al., 2008). Or an R list, a Python dictionary or even a C structure, see (Matloff, 2011)

a prediction, a loss function to measure the distance between the prediction and the true label (which is a loss score used as a feedback signal), an optimizer to adjust the weights in order to reduce the loss score in the next instance of the input of features. As a result, the model learns with many layers of data transformation. All the previous shallow learning has one layer where its weights are learned by the classifier. For instance, in our base-line, the Logistic Regression is applied to the training data, and the then calculated weights are used to predict the label, and then a loss function permits finding the accuracy of the test. In shallow learning, as with the LR, the process still includes a feedback signal to fine-tune the weights since each data point permits a more specific adjustment of the logistic curve to the data, but this happens in a one-layer-deep circuit. In DL the architecture of the data transformation is made as complex as needed so that the fine-tuning is through a multi-dimensional chain of data learning processes; learning is embedded in many layers of discretional data munging. Using deep learning, it is possible to produce the best performance for the problem at hand with an accuracy of 96% after transforming the features to tensors, coding categorical variables as 2D-tensors with normalization, and building a sequential neural network with two Dense layers of sixteen hidden units, a "relu" (rectified linear unit, or non-linearity) activation, plus a final layer with just one hidden unit and a "sigmoid" activation. We use the Keras library, and Tensor Flow backend (Abadi et al., 2015) Notice that this model does not use the power of the OccStrings analyzed in the previous section. So, the improvements can only be attributable to the DL method. Figure 8 shows the loss and the accuracy of both the training and the validation sets. Overall the model performs well both in the training and, importantly, also in the validation sets. This implies that the model *generalizes* well since it performs well on data it has never seen before. Also, it suggests that *overfitting* to the training data is not a problem for this model with its current *capacity* (or number of learnable parameters) and amount of training data. This result permits us to conclude that DL outperforms the conventional ML models in the task of binary classification when general features are included (but not OccString)

6.5.2. Deep Learning for text-data

One interesting question to be answered in this paper is whether a DL model could outperform the very successful 1851 Bag of word conventional ML model with AdaBoost and using a pre-processed bag of words vector encoding. To test this we built two models as described in the Methodology: one SNN with Embedding, that is a dense, lower-dimensional and learned from data word representation in contrast with the sparse, high-dimensional, and hardcoded bag of word representation; the other a RNN model with a LSTM layer. Both perform better than the 1851 Bag of word with accuracies of 0.9964, and 0.9978 compared with 0.9949. The LSTM loss and accuracy are shown in Figure 9.

7. Assessment and Conclusion

This paper uses methodological advances in machine learning to apply to historical census data classification. In particular, it has shown that boosting, followed by ensemble methods, sometimes associated with LR, among the probabilistic approaches generate sizable improvements in accuracy over the benchmark of a stand-alone LR for the classification of individuals by their entrepreneurial status for the early censuses. The results tend to confirm Hindman (2015) who suggests that "Ensemble models illustrate what is possible in terms of predictive accuracy, and they provide the best yardstick with which to judge simpler models" ... "Ensemble methods .... are almost always a superior choice to the OLS and logit models that dominate empirical social science work today". At the same time, significant improvements in the accuracy of the census classifications assessed here can be achieved with a bag-of-words strategy using the OccString feature in the data, which employs advances in text and natural language pre-processing. However, DL with neural networks performs at the top of all the tested models. This confirms that ML, and, especially, DL can be actively developed to tackle classification of historical data. This case study has proved that ML and, in particular, DL are techniques that are valuable for classification of historical data; they also encourage subsequent exploration of record linkage of historical census data as recently described by Capobianco and Marinai (2019) and Liu et al. (2019). Our efforts demonstrate that a multidisciplinary approach to traditional information classification tasks can realize the potential of a "big data revolution" (Hindman, 2015). As Hindman (2015) suggests "[n]ew data sources and better algorithms do allow social scientists of all stripes to offer most accurate forecasts in many ... areas". Finally, the addition of machine learning to traditional methodological techniques suggests that the use of big data techniques (even for small sample queries) can help to understand and improve testing of theory.

The main implication of our results is to show that expanding the range of methods applied to the binary classification of big data in the information sciences can produce important increases in performance so researchers should actively look for the best performing algorithm. Our results highlight that theoretically binary classification tasks like the ones presented in the empirical part of this paper, should not only be tackled by standard LR but also should explore more general ML algorithms, especially boosting and ensemble methods, which have performances that greatly outperform the LR. The paper suggests that a wide range of ML methods are preferable options to solve classification tasks. As our results show ten common but optimized, ML algorithms perform better than the LR in almost every case. This is a valuable conclusion indicating that researchers should test and compare different ML algorithms before accepting the results from any one method. This should not be surprising since it confirms Wolpert's (1996) famous "no free lunch theorem" (i.e. that there is no universally best model). Also, our results show that empirically, AdaBoost outperformed all other ML methods. At the same time, DL as a special method and in particular using TensorFlow library is an even better choice for our data, indicating that future research should focus on further developments of neural network algorithms as classifier tools for this task.

Another lesson from this paper is that text-based classifications perform better in both shallow and deep learning. Similar methodologies in the literature were interpreted by Kastrati et al. (2019) as integrating learning into an ontology-based on the semantics in the text, or by Abdi et al. (2019) as allowing sentiment analysis, although our text descriptors are too brief and simplistic to utilize such approaches. However, shallow models' text-encoding as a bag of word and deep models' text-embedding perform at the top of the list among all the model tested. This suggests avenues for future research.

Overall, our efforts have shown that the most accurate method for the task at hand is Deep Learning both if we restrict to using or not using OccString. Deep Learning outperforms AdaBoost when all the features but OccString are used, and Deep Learning outperforms the 1851 Bag-of-words AdaBoost model especially with an RNN architecture. The Deep Learning model with LSTM layer reaches a 0.9978 accuracy, and the top-ranked accuracy among all our models. Thus, our recommendations are to use Deep Learning and use OccString when this variable is available (as in the earlier censuses 1851-1881).

The paper has focused on the methodological advances offered by ML, and the comparison of different methods for implementing ML. Future developments of these methods can be used to join up the historical censuses with modern data so that the long-term trends in entrepreneurship can be examined over time. This begins to allow evaluations of the effects of changing descriptors of entrepreneurial behaviour and the effect of different economic conditions on decision choices between waged work and employer or own account status. Indeed the results of the application of the methods used here for identifying entrepreneurial status 1851-81 linked to the later period 1891-1911, and then linked to modern censuses show that the Victorian period had a higher rate of entrepreneurship than any subsequent time in Britain (Bennett et al., 2019). Besides, the availability of a database on the full population of entrepreneurs over time can be used to study the statistical characteristics of the firm size distribution (Montebruno et al., 2019b,c), and the study of the determinants of Victorian entrepreneurship (Bennett et al., 2019). Moreover the data deposit of the estimates of entrepreneurial status based on the methods used in this paper allow other researchers to develop answers to other research questions; such as persistence in entrepreneurship over time, growth and change in firm sizes, and using record-linkage between census years, open up new potential to examine the life stages and career evolution of entrepreneurs and switching between different employment statuses.

Acknowledgments

This research has been supported by the ESRC under project grant ES/M010953: **‘Drivers of Entrepreneurship and Small Businesses’** and Isaac Newton Trust research grant 17.07(d): **‘Business Employers in 1871’**.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Computer software.

Abdi, A., Shamsuddin, S.M., Hasan, S., Piran, J., 2019. Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion. *Information Processing and Management* 56, 1245–1259. doi:10.1016/j.ipm.2019.02.018.

Al-Salemi, B., Ayob, M., Kendall, G., Noah, S.A.M., 2019. Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms. *Information Processing and Management* 56, 212–227. doi:10.1016/j.ipm.2018.09.008.

Alvarez-Galvez, J., 2016. Discovering complex interrelationships between socioeconomic status and health in Europe: A case study applying Bayesian Networks. *Social Science Research* 56, 133–143. doi:10.1016/j.ssresearch.2015.12.011.

Bennett, R.J., Smith, H., van Lieshout, C., Montebruno, P., Newton, G., 2019. *The Age of Entrepreneurship: Business Proprietors, Self-employment and Corporations Since 1851.* Routledge international studies in business history, London and New York. doi:10.4324/9781315160375.

Bennett, R.J., Smith, H., Montebruno, P., 2018. The Population of Non-corporate Business Proprietors in England and Wales 1891–1911. *Business History* doi:10.1080/00076791.2018.1534959.

Blanchflower, D.G., Oswald, A.J., 1998. What Makes an Entrepreneur? *Journal of Labor Economics* 16, 26–60. doi:10.3386/w3252.

Boutell, M.R., Luo, J., Shen, X., Brown, C.M., 2004. Learning multi-label scene classification. *Pattern Recognition* 37, 1757–1771. doi:10.1016/j.patcog.2004.03.009.

Cameron, A.C., Trivedi, P.K., 2005. *Microeconometrics: methods and applications.* Cambridge University Press, Cambridge. doi:10.1017/CBO9780511811241.

Capobianco, S., Marinai, S., 2019. Deep neural networks for record counting in historical handwritten documents. *Pattern Recognition Letters* 119, 103–111. doi:10.1016/j.patrec.2017.10.023.

Cheng, W., Hüllermeier, E., 2009. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning* 76, 211–225. doi:/10.1007/s10994-009-5127-5.

Chollet, F., 2018. *Deep learning with Python.* Manning, Shelter Island, New York.

Dawe, N., 2018. Python Code for Two-class AdaBoost. Computer software (3-clause BSD License).

Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 861–874. doi:10.1016/j.patrec.2005.10.010.

Freund, Y., Schapire, R.R.E., 1996. Experiments with a New Boosting Algorithm. *International Conference on Machine Learning* , 148–156doi:10.1.1.133.1040.

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 1189–1232.

Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., Brinker, K., 2008. Multilabel classification via calibrated label ranking. *Machine Learning* 73, 133–153. doi:10.1007/s10994-008-5064-8.

Géron, A., 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* O'Reilly, Sebastopol, California.

Goldberger, A.S., 1991. *A course in econometrics.* Harvard University Press, Cambridge, Massachusetts; London, England.

Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep learning.* Adaptive computation and machine learning, The MIT Press, Cambridge, Massachusetts; London, England.

Head, T., 2018. Python Code for ROC curve. Computer software (3-clause BSD License).

Higgs, E., 2004. *Life, death and statistics: civil registration, censuses and the work of the General Register Office, 1836-1952.* Local population studies supplement, Local Population Studies, Hatfield, England.

Higgs, E., Schürer, K., 2014. Integrated Census Microdata (I-CeM), 1851-1911, UK Data Archive data deposit SN-7481. doi:10.5255/UKDA-SN-7481-1.

Hindman, M., 2015. Building Better Models: Prediction, Replication, and Machine Learning in the Social Sciences. *The ANNALS of the American Academy of Political and Social Science* 659, 48–62. doi:10.1177/0002716215570279.

Hu, Y., Li, H., Cao, Y., Li, T., 2006. Automatic extraction of titles from general documents using machine learning. *Information Processing & Management* 42, 1276–1293. doi:10.1016/j.ipm.2005.12.001.

Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K., 2008. Label ranking by learning pairwise preferences. *Artificial Intelligence* 172, 1897–1916. doi:10.1016/j.artint.2008.08.002.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. *An introduction to statistical learning: with applications in R.* Springer texts in statistics, Springer, New York; Heidelberg; Dordrecht; London.

Kastrati, Z., Imran, A.S., Yayilgan, S.Y., 2019. The impact of deep learning on document classification using semantically rich representations. *Information Processing and Management* 56, 1618–1632. doi:10.1016/j.ipm.2019.05.003.

Kucukyilmaz, T., Cambazoglu, B.B., Aykanat, C., Baeza-Yates, R., 2017. A machine learning approach for result caching in web search engines. *Information Processing and Management* 53, 834–850. doi:10.1016/j.ipm.2017.02.006.

Liu, Y., Jin, L., Lai, S., 2019. Automatic labeling of large amounts of handwritten characters with gate-guided dynamic deep learning. *Pattern Recognition Letters* 119, 94–102. doi:10.1016/j.patrec.2017.09.042.

Matloff, N.S., 2011. *The art of R programming.* No Starch Press, San Francisco, California.

McCulloch, W., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5, 115–133. doi:10.1007/bf02478259.

Montebruno, P., Bennett, R., Van Lieshout, C., Smith, H., Satchell, A., 2019a. Shifts in agrarian entrepreneurship in mid-Victorian England and Wales. *The Agricultural History Review* 67, 71–108.

Montebruno, P., Bennett, R.J., van Lieshout, C., Smith, H., 2019b. A tale of two tails: Do Power Law and Lognormal models fit firm-size distributions in the mid-Victorian era? *Physica A: Statistical Mechanics and its Applications* 573, 858–875. doi:10.1016/j.physa.2019.02.054.

Montebruno, P., Bennett, R.J., van Lieshout, C., Smith, H., 2019c. Research data supporting "A tale of two tails: Do Power Law and Lognormal models fit firm-size distributions in the mid-Victorian era?". *Mendeley Data* doi:10.17632/86xkkncmw3.1.

Montebruno, P., Bennett, R.J., Smith, H., van Lieshout, C., 2020. Research data supporting "Machine learning in the processing of historical census data". *Mendeley Data* doi:10.17632/p4zptr98dh.1.

Murphy, K., 2012. *Machine Learning A Probabilistic Perspective.* The MIT Press, Cambridge, Massachusetts, London, England.

Murthy, D., Gross, A.J., 2017. Social media processes in disasters: Implications of emergent technology use. *Social Science Research* 63, 356–370. doi:10.1016/j.ssresearch.2016.09.015.

Parker, S.C., 2004. *The Economics of Self-Employment and Entrepreneurship.* Cambridge University Press, Cambridge, England. doi:10.1017/cbo9780511493430.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825–2830.

Rabe-Hesketh, S., Skrondal, A., 2012. *Multilevel and longitudinal modelling using Stata. Volume 2, Categorical responses, counts, and survival.* 3rd ed. ed., Stata Press, College Station, Texas.

Read, J., Pfahringer, B., Holmes, G., Frank, E., 2011. Classifier chains for multi-label classification. *Machine Learning* 85, 333–359. doi:10.1007/s10994-011-5256-5.

Reichenberg, O., Berglund, T., 2019. "Stepping up or stepping down?": The earnings differences associated with Swedish temporary workers' employment sequences. *Social Science Research* doi:10.1016/j.ssresearch.2019.04.007.

Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 386–408. doi:10.1037/h0042519.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning Internal Representations by Error Propagation, in: Collins, A., Smith, E.E. (Eds.), Readings in Cognitive Science. Morgan Kaufmann, pp. 399–421. doi:10.1016/B978-1-4832-1446-7.50035-2.

Schapire, R., Singer, Y., 1999. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning* 37, 297–336. doi:10.1023/A:100761452.

Schürer, K., Penkova, T., Shi, Y., 2015. Standardising and Coding Birthplace Strings and Occupational Titles in the British Censuses of 1851 to 1911. *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 48, 195–213. doi:10.1080/01615440.2015.1010028.

Schwartz, R.L., Phoenix, T., Foy, b.d., 2008. *Learning Perl*. 5th ed., O'Reilly, Beijing; Farnham.

Su, Z., Meng, T., 2016. Selective responsiveness: Online public demands and government responsiveness in authoritarian China. *Social Science Research* 59, 52–67. doi:10.1016/j.ssresearch.2016.04.017.

Tang, B., Kay, S., He, H., 2016. Toward Optimal Feature Selection in Naive Bayes for Text Categorization. *IEEE Transactions on Knowledge and Data Engineering* 28, 2508–2521. doi:10.1109/tkde.2016.2563436.

Tang, X., Chen, L., Cui, J., Wei, B., 2019. Knowledge representation learning with entity descriptions, hierarchical types, and textual relations. *Information Processing and Management* 56, 809–822. doi:10.1016/j.ipm.2019.01.005.

The Editors of the American Heritage Dictionaries, 2011. *The American Heritage dictionary of the English language*. Houghton Mifflin Harcourt, Boston.

Tong, S., Chang, E., 2001. Support vector machine active learning for image retrieval, in: Proceedings of the ninth ACM international conference on multimedia, ACM. pp. 107–118. doi:10.1145/500156.500159.

Tsoumakas, G., Katakis, I., Vlahavas, I., 2011. Random k-Labelsets for Multilabel Classification. *IEEE Transactions on Knowledge and Data Engineering* 23, 1079–1089. doi:10.1109/tkde.2010.164.

UK Parliament, 1890. *The Census. Report of the Committee appointed by the Treasury to inquire into certain questions connected with the taking of the Census, presented to both Houses of Parliament by Command of Her Majesty. LVIII.13.* Printed for Her Majesty Stationery Office by Eyre and Spottiswoode, London, England.

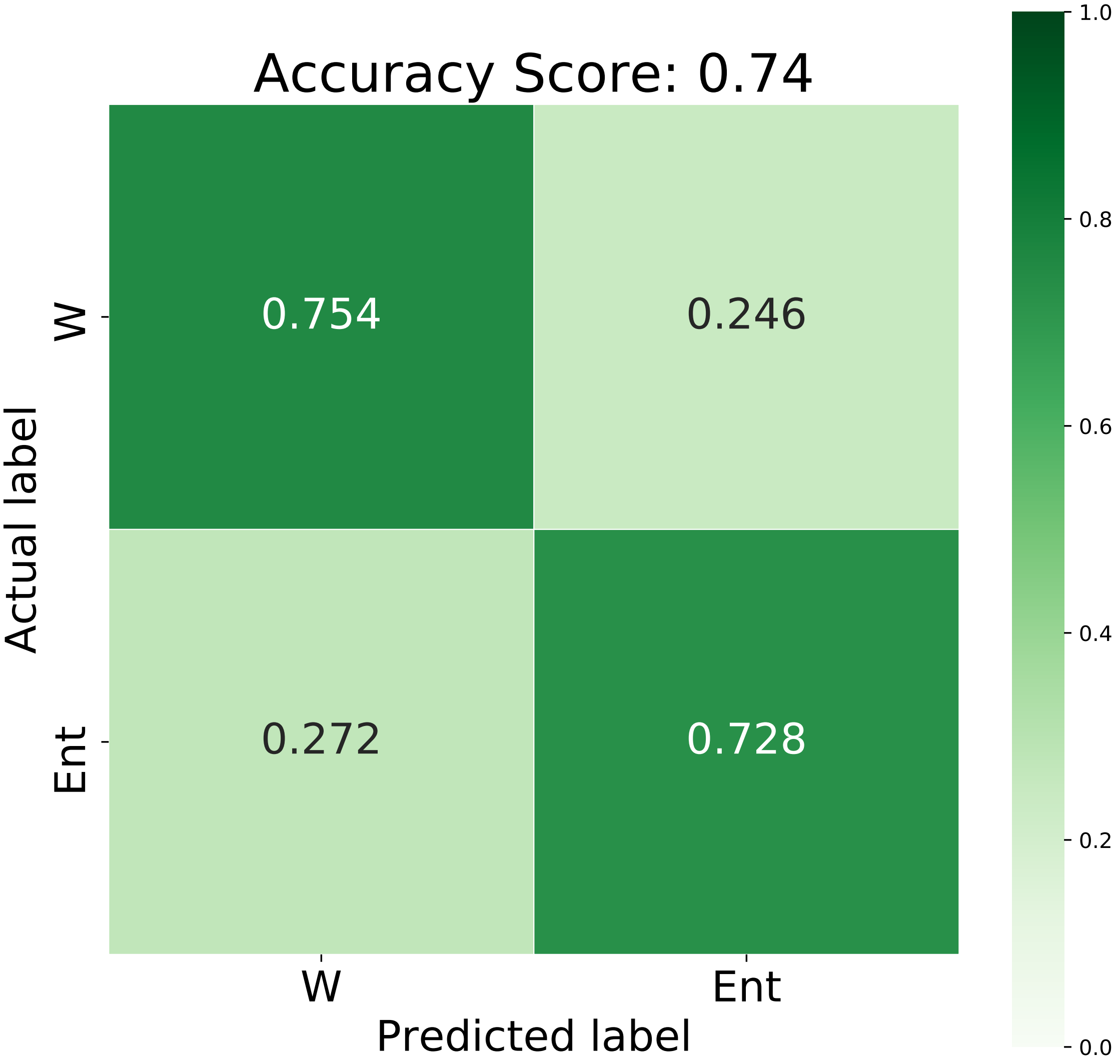
Varoquaux, G., Müller, A., 2018. Python Code for Classifier Comparison. Modified for documentation by Jaques Grobler. Computer software (3-clause BSD License).

Wolpert, D.H., 1996. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation* 8, 1341–1390. doi:10.1162/neco.1996.8.7.1341.

Wu, Q., Ye, Y., Zhang, H., Ng, M.K., Ho, S.S., 2014. ForesTexter: An efficient random forest algorithm for imbalanced text categorization. *Knowledge-Based Systems* 67, 105–116. doi:10.1016/j.knosys.2014.06.004.

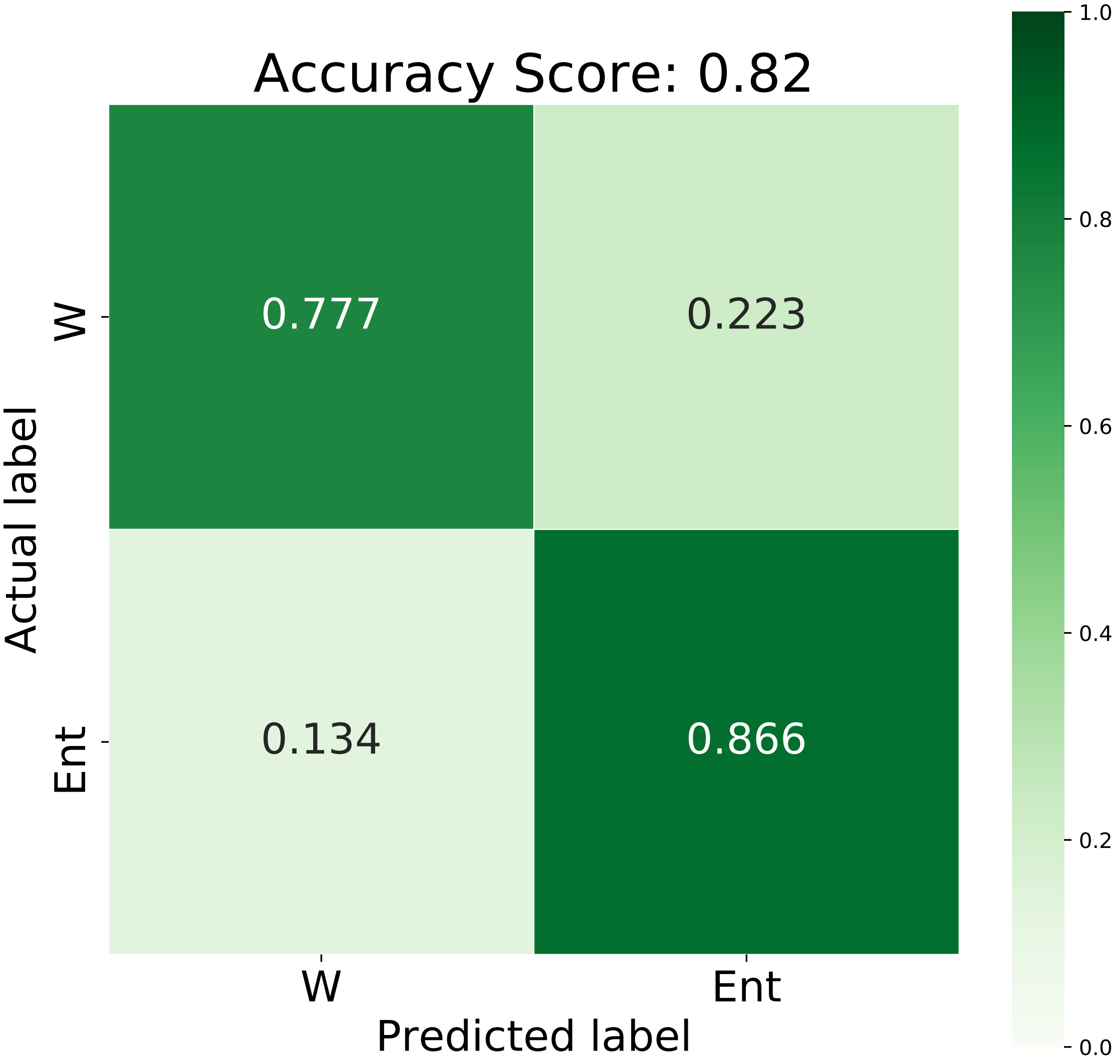
Zhang, M.L., Zhou, Z.H., 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition* 40, 2038–2048. doi:10.1016/j.patcog.2006.12.019.

Figure 1: Confusion matrix for the binary classification of being W or Ent: 1891 Logistic Regression.



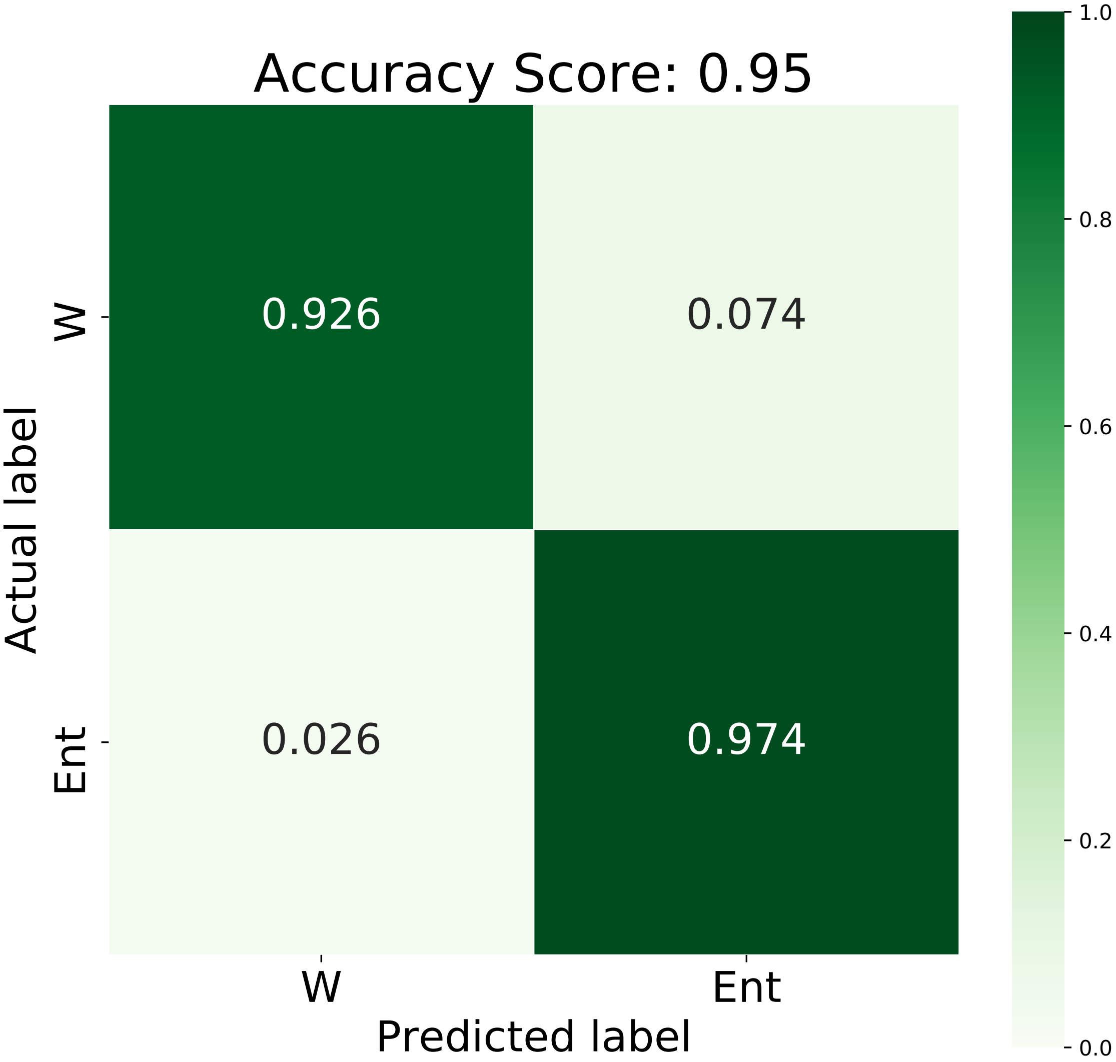
Input data 1,000 W / 1,000 Ent

Figure 2: Confusion matrix for the binary classification of being W or Ent: 1851 Logistic Regression



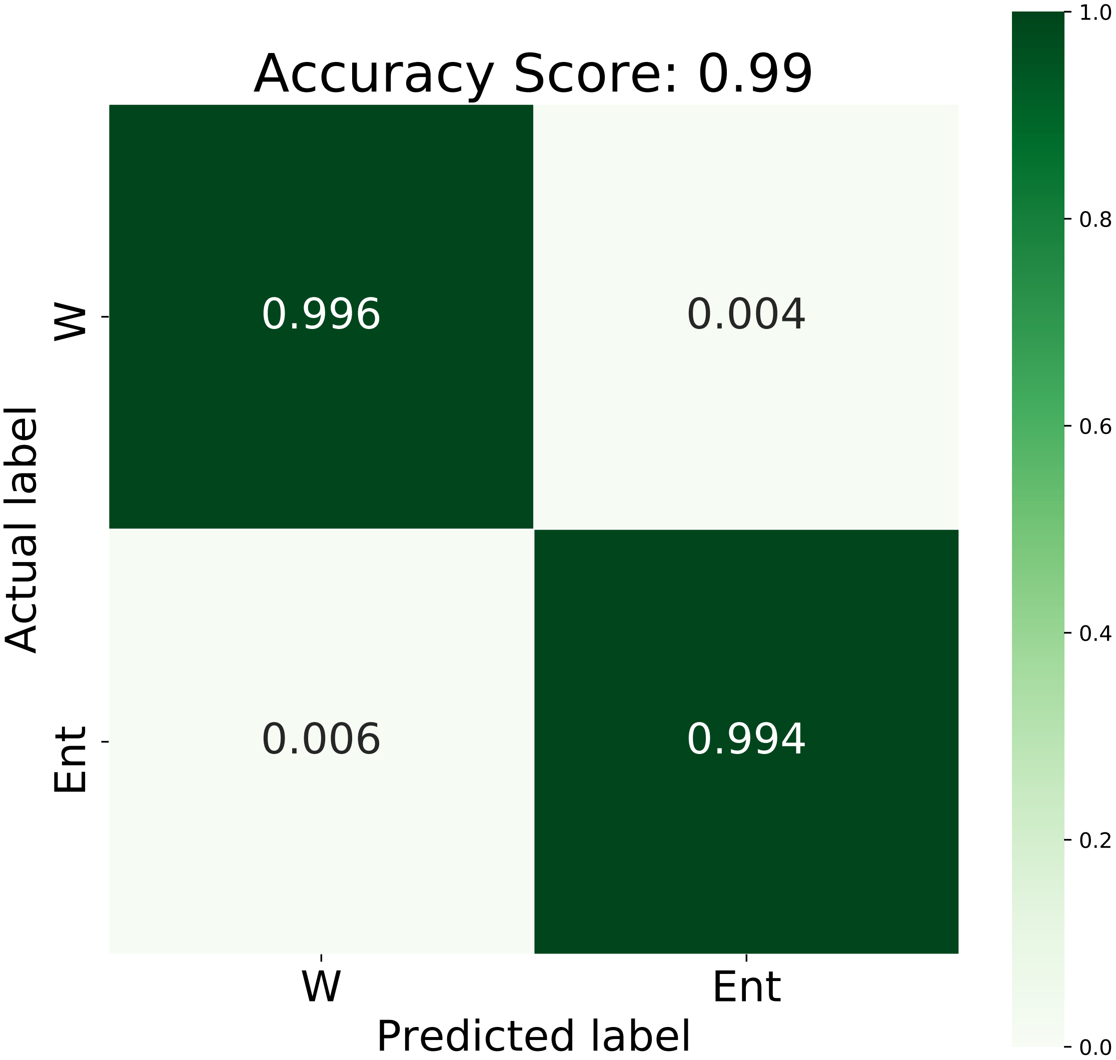
Input data 1,000 W / 1,000 Ent

Figure 3: Confusion matrix for the binary classification of being W or Ent: 1851 AdaBoost



Input data 70,872 W / 70,872 Ent

Figure 4: Confusion matrix for the binary classification of being W or Ent using the AdaBoost classifiers with the OccString feature (Bags of words), 1851



Input data 70,872 W / 70,872 Ent

Figure 5: Ents and Ws classification: True Positive, False Positive, False Negative, True Negative

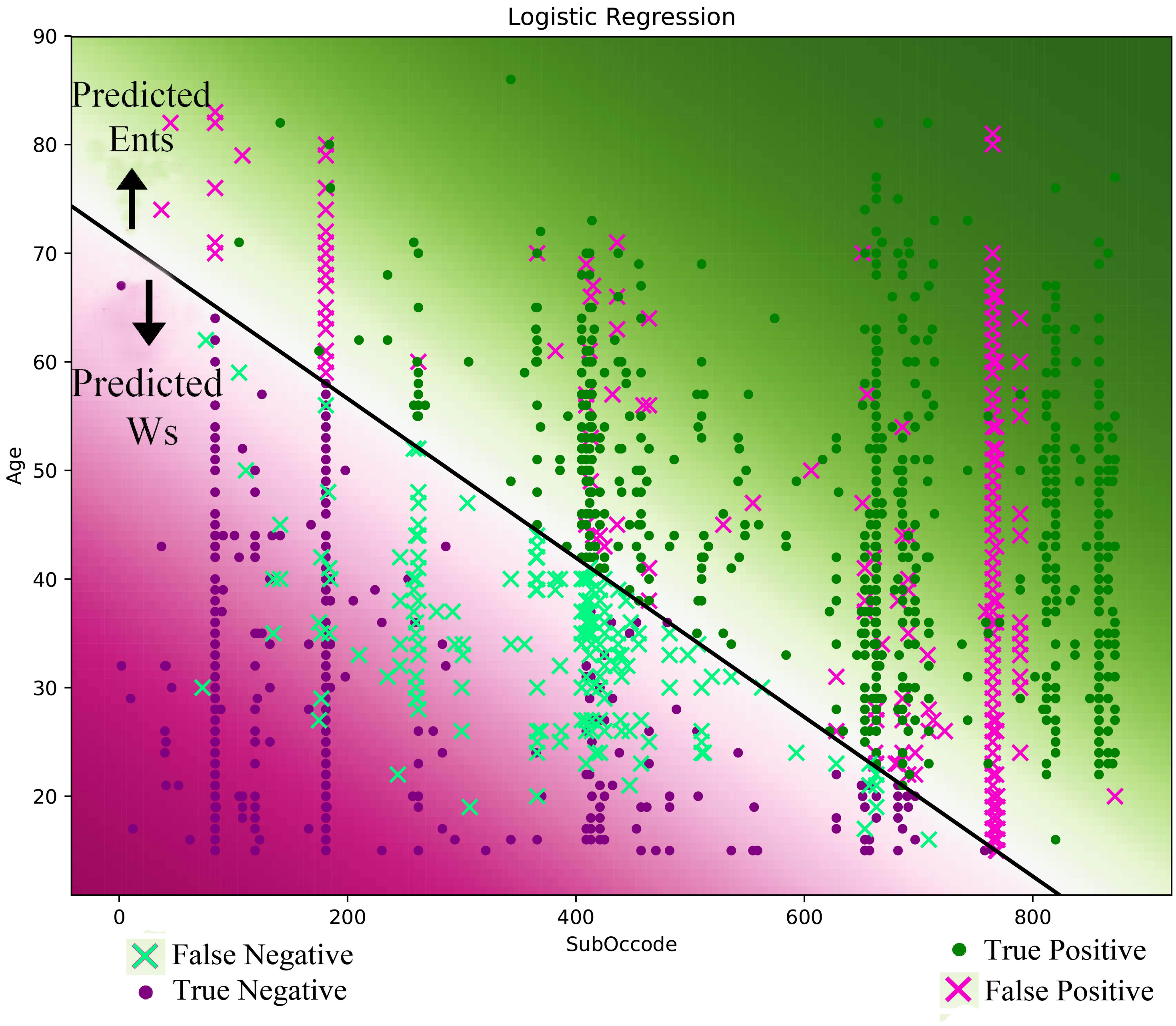
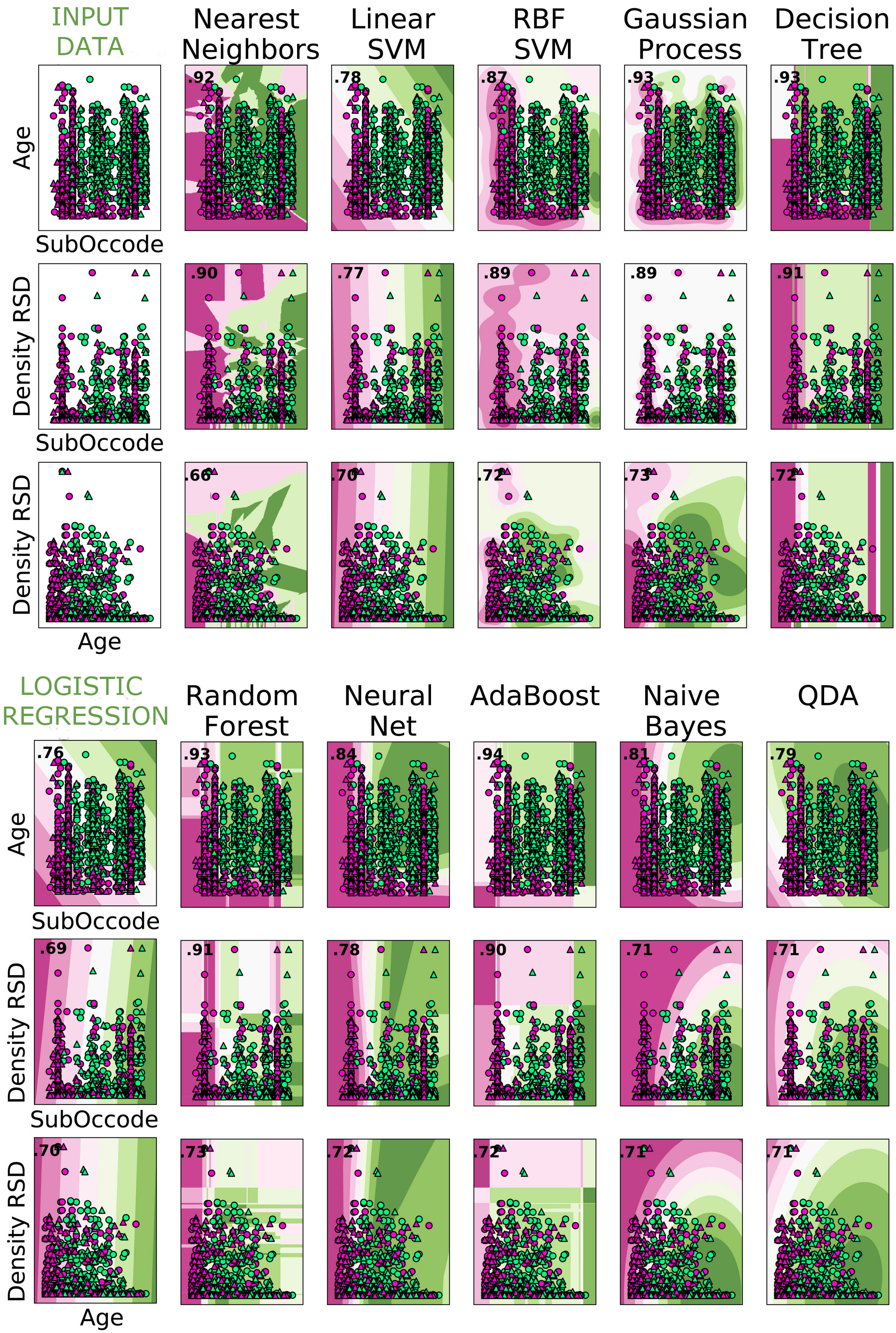


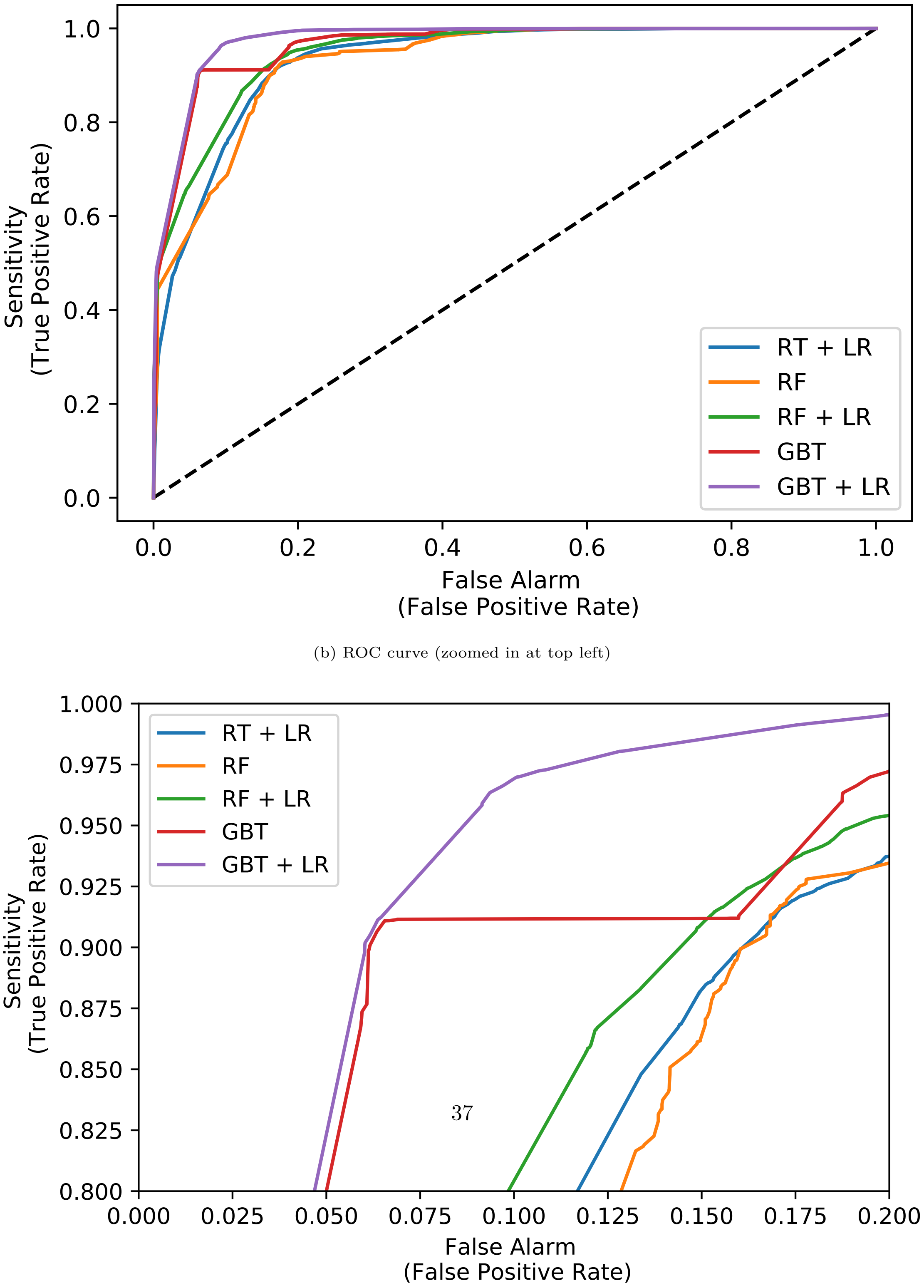
Figure 6: 2-D predicted probability and accuracy comparing Logistic Regression to competing classification algorithms



Note to *Figure 5* 2-D predicted probability and accuracy comparing Logistic Regression (LR, three boxes down and far left) to ten alternative and competing classification algorithms (Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes, and QDA) for the label being an Ent with 2000 balanced random data points (1000 Ws, 500 Es and 500 OAs. That is 1000 Ws and 1000 Ents). The three boxes (the upper using SubOccode and Age, the middle using SubOccode and RSD Density, and the bottom using Age and RSD Density) in the first half-column, i.e., up and far left are the input data. The purple figures—that is circles and triangles—are Ws and the green ones are Ents. The circles are the training set (60% of the total) and the triangles are the testing set (40% of the total). The input data are repeated in each classification algorithm with the background color being the 2-D predicted probability grid: when the grid is purple a test point—that is a triangle—is classified as W irrespective of its true value or color and when the color is green a test point is classified as Ent, also irrespective of its true value or color. According to this classification of the test points the accuracy for each method is calculated. The code is used from (Varoquaux and Müller, 2018) under a 3-clause BSD License.

Figure 7: Receiving operating charateristic (ROC) curve for the binary classification of being or not an Entrepreneur using RandomTrees (RT), RandomForest (RF), GradientBoosting (GBT) as stand-alone methods or combined with Logistic Regression (LR)

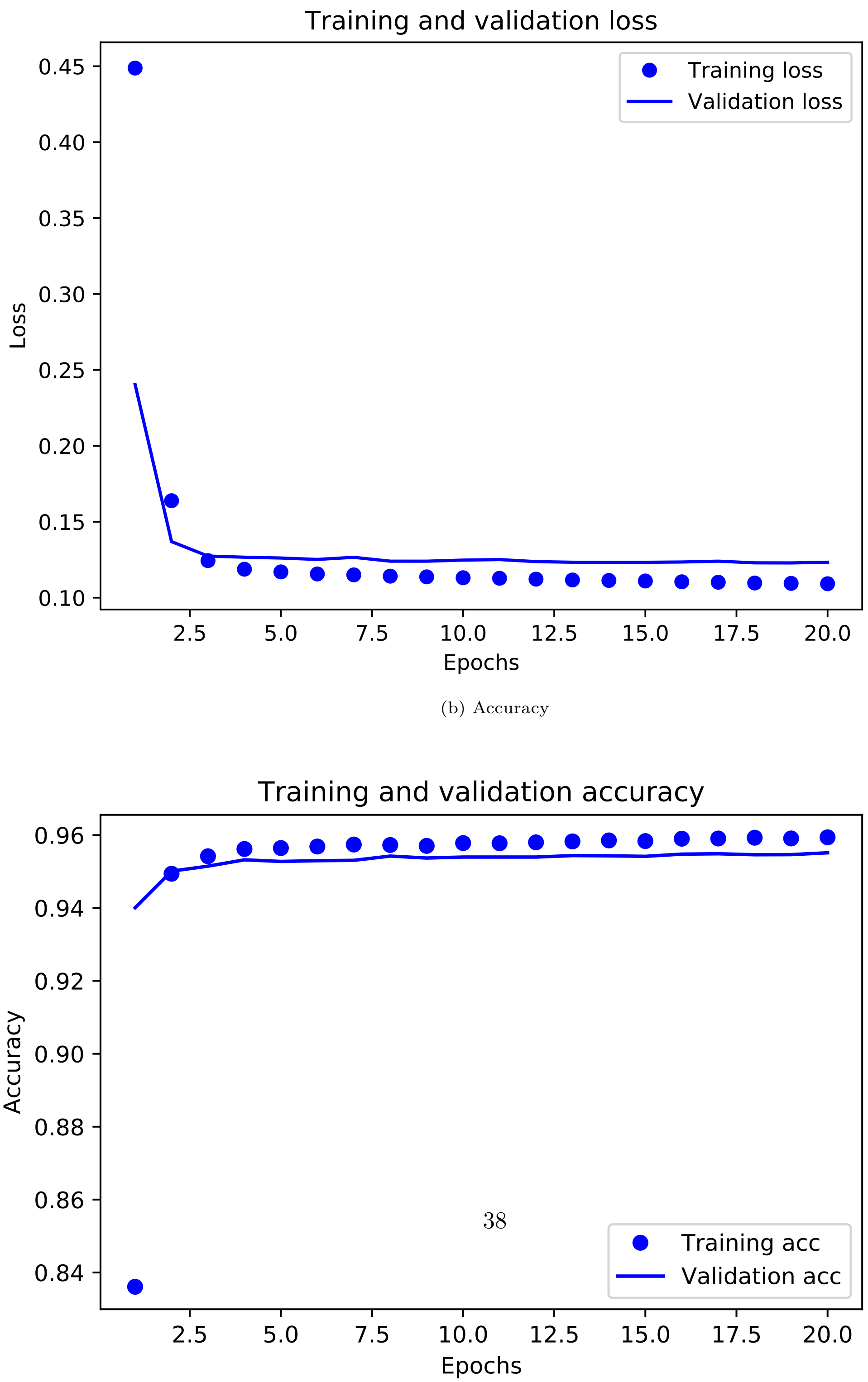
(a) ROC curve



(b) ROC curve (zoomed in at top left)

Input data 70,872 W / 70,872 Ent

Figure 8: Loss and accuracy of deep learning for a Dense Sequential Neural Network (SNN) with all the features but not with OccString in the maximum possible set of extracted Entrepreneurs



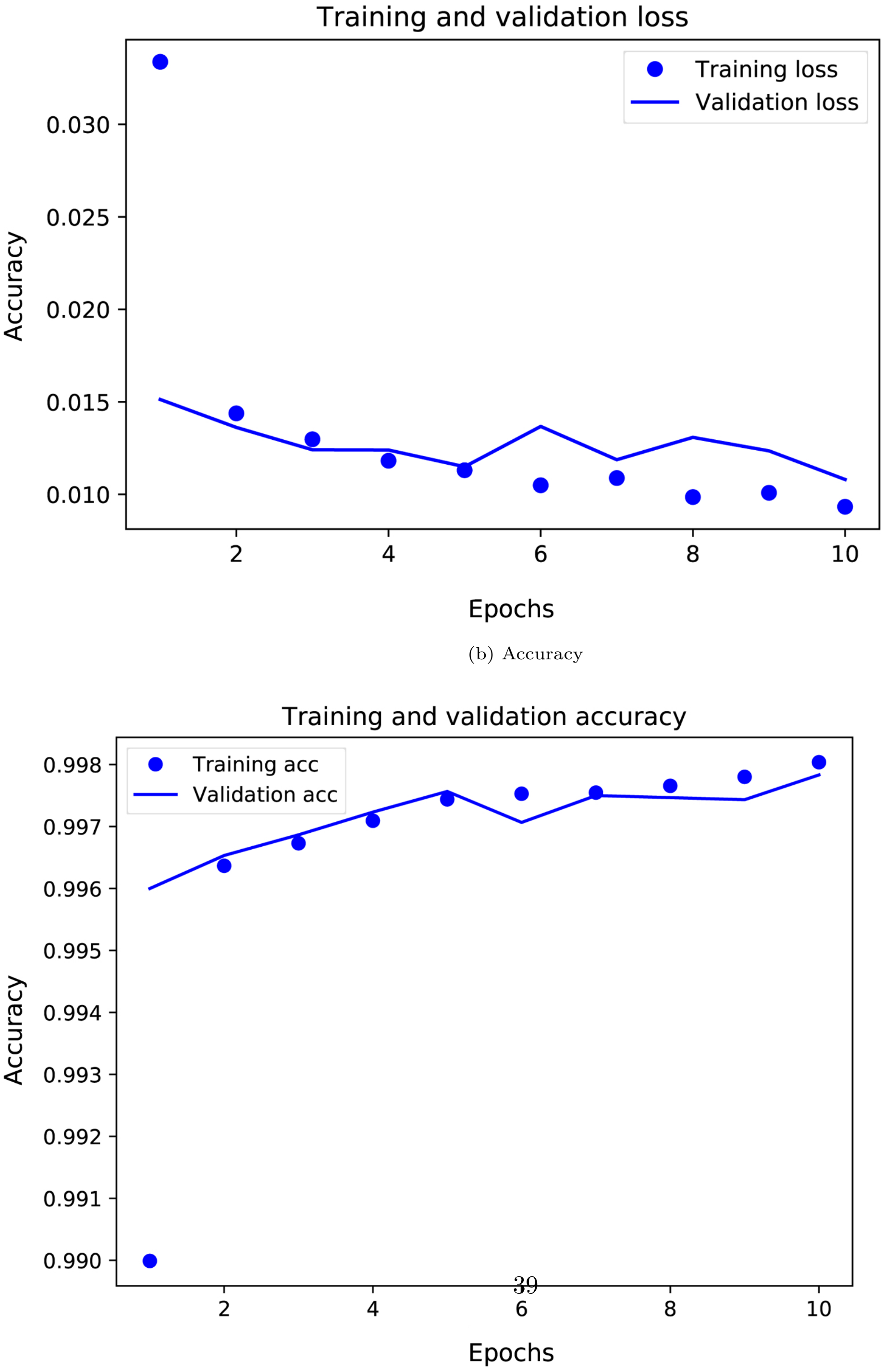
(a) Loss

(b) Accuracy

Input data 70,872 Workers, 35,436 Employers, and 35,436 Own accounts. Maximum possible set of extracted: that is labeled according to their strings or type of employment occupation codes. After farmers have been dropped from the set.

Figure 9: Loss and accuracy of deep learning for the Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) with only OccString feature in the maximum possible set of extracted Entrepreneurs

(a) Loss



(b) Accuracy

Input data as for Figure 8.

Table 1: 1891 census training set features for binary labels: Worker (W) / Entrepreneur (Ent).

	Means				Medians				Overall	
	Worker	Entrepreneur	t	p-value	Worker	Entrepreneur	z	p-value	min	max
Age	32.1	44.6	-976	0	28	42	-941	0	15	90
RSD Density	28.3	25.9	53.7	0	2.84	1.91	166	0	0	392
Sex	1.3	1.28	60	0	1	1	60	0	1	2
Marital status	1.57	2.07	-737	0	1	2	-767	0	1	4
Relationship to the head	2.97	1.61	627	0	2	1	813	0	1	10
Number of Servants	.0785	.432	-708	0	0	0	-847	0	0	99
SubOccode	344	487	-652	0	248	482	-589	0	1	882

The table shows the means, the t-value and p-value of the two-sample t-test with equal variances and the medians, the z-value and p-value of the two-sample Wilcoxon rank-sum, or Mann-Whitney, test for the the seven features of the training set. Sex is coded as 1 Male (base category), and 2 Female. Marital status as 1 Single (base category), 2 Married, 4 Widowed. Relationship to the head as 1 Head (base category), 2 Conjugal Family Unit (CFU) member, 3 Older generation, 4 Sibling, 5 Other family, 6 Servant, 7 Working title, 8 Lodger/boarder, 9 Non-household, 10 Unknown. RSD Density is the Registration SubDistrict Density. SubOccode is coded 1 to 882 (with gaps) with SubOccode 196. Coal Miners - Hewers, Workers at the Coal Face as base category. (See Bennett et al. (2018) for a full list of the 844 categories of SubOccode)

Table 2: 1891 census training set features for multi-class labels: Worker (W) / Employer (E) / Own account (OA).

	Means								
				W/E		W/OA		E/OA	
	Worker	Employer	Own account	t	p-value	t	p-value	t	p-value
Age	32.1	45.9	43.9	-676	0	-753	0	79.1	0
RSD Density	28.3	22	28.1	89	0	3.5	.000445	-71.8	0
Sex	1.3	1.11	1.37	303	0	-142	0	-358	0
Marital status	1.57	2.07	2.08	-467	0	-607	0	-7.4	1.36e-13
Relationship to the head	2.97	1.34	1.75	459	0	450	0	-149	0
Number of Servants	.0785	.871	.184	-1,021	0	-207	0	428	0
SubOccode	344	435	517	-261	0	-646	0	-182	0

	Medians								
				W/E		W/OA		E/OA	
	Worker	Employer	Own account	z	p-value	z	p-value	t	p-value
Age	28	45	43	-653	0	-725	0	79.8	0
RSD Density	2.84	.866	2.26	211	0	58.4	0	-136	0
Sex	1	1	1	302	0	-142	0	-344	0
Marital status	1	2	2	-563	0	-573	0	65.2	0
Relationship to the head	2	1	1	635	0	568	0	-233	0
Number of Servants	0	0	0	-1,115	0	-334	0	443	0
SubOccode	248	409	657	-256	0	-566	0	-159	0

The table shows the means, the t-value and p-value of the two-sample t-test with equal variances and the medians, the z-value and p-value of the two-sample Wilcoxon rank-sum, or Mann-Whitney, test for the the seven features of the training set. Data defined in Table 1.

Table 3: Logit model weights.

Feature	Labels: W/Ent Weight ($\boldsymbol{w}$)	 t-stat
SubOccode		
52. Schoolmasters And Teachers (Default) Minus Suboccode 802	3.959***	(126.62)
105. Laundry Wrk: Washer, Iron, Etc. (Not Dom) Minus Suboccode 805	5.038***	(167.28)
141. Carmen Carriers Carters And Draymen	3.622***	(118.78)
173. Farmer, Grazier	7.342***	(242.39)
196. Coal Miners - Hewers, Workers At The Coal Face	0	(.)
262. Blacksmiths Minus Suboccode 812	3.861***	(127.46)
409. Carpenter, Joiner Minus Suboccode 820	3.601***	(120.29)
551. Cotton & Cotton Good Mf Weaving Processes	0.136*	(2.26)
653. Tailors Not Merchants- Default Minus Subocc 858	4.482***	(149.15)
657. Dressmakers	6.768***	(226.57)
663. Shoe & Boot Maker (& Repairer) Minus Suboccode 862	4.721***	(159.43)
RSD Density	-0.00706***	(-132.86)
RSD Density × RSD Density	0.0000175***	(79.98)
Age	0.135***	(239.21)
Age × Age	-0.00102***	(-165.09)
Sex		
1. Male	0	(.)
2. Female	-0.0363***	(-5.58)
Marital status		
1. Single	0	(.)
2. Married	-0.106***	(-17.45)
4. Widowed	-0.0167	(-1.95)
2. Female × 2. Married	0.297***	(33.05)
2. Female × 4. Widowed	0.0300**	(2.89)
Relationship to the head		
1. Head	0	(.)
2. CFU member	-0.829***	(-138.86)
3. Older generation	-0.893***	(-52.83)
4. Siblings	-0.728***	(-72.65)
5. Other family	-1.053***	(-77.18)
6. Servants	-3.186***	(-69.34)
7. Working title	-2.773***	(-75.91)
8. Lodgers/boarders	-1.184***	(-162.58)
9. Non-household	-1.429***	(-48.37)
10. Unknown	-0.574***	(-46.14)
Number of servants	0.524***	(154.61)
Constant	-8.851***	(-278.30)
Observations	7,213,217	

t statistics in parentheses

*** $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$**

Weights ($\boldsymbol{w}$, second column) calculated from the 1891 training data for features (base category) SubOccode (196) (Only 11 example SubOccodes are shown from the 844 estimated), RSD Density, Age, Sex (Male), Marital status (Single), Relationship to the head (Head), and Number of servants. Binary labels are Worker (W) and Entrepreneur (Ent). In parentheses, $\boldsymbol{t}$-statistics (third column).

Table 4: Logit model weights and marginal effects without the SubOccode.

	Labels: W/Ent	
Feature	$\boldsymbol{w}$**/se**	$\boldsymbol{\partial y/\partial x}$**/se**
RSD Density	-0.005***	-0.001***
	(0.000)	(0.000)
RSD Density × RSD Density	0.000***	
	(0.000)	
Age	0.115***	0.009***
	(0.000)	(0.000)
Age × Age	-0.001***	
	(0.000)	
Sex		
1. Male	0.000	0.000
	(.)	(0.000)
2. Female	0.776***	0.143***
	(0.004)	(0.001)
Martial status		
1. Single	0.000	0.000
	(.)	(0.000)
2. Married	-0.286***	-0.036***
	(0.005)	(0.001)
4. Widowed	-0.268***	-0.039***
	(0.007)	(0.001)
2. Female × 2. Married	0.174***	
	(0.007)	
2. Female × 4. Widowed	0.011	
	(0.008)	
Relationship to the head		
1. Head	0.000	0.000
	(.)	(0.000)
2. CFU member	-0.902***	-0.138***
	(0.005)	(0.001)
3. Older generation	-1.128***	-0.162***
	(0.013)	(0.001)
4. Siblings	-0.807***	-0.127***
	(0.008)	(0.001)
5. Other family	-1.071***	-0.156***
	(0.013)	(0.001)
6. Servants	-3.240***	-0.253***
	(0.045)	(0.001)
7. Working title	-2.256***	-0.230***
	(0.035)	(0.001)
8. Lodgers/boarders	-1.208***	-0.169***
	(0.006)	(0.001)
9. Non-household	-1.549***	-0.195***
	(0.025)	(0.002)
10. Unknown	-0.232***	-0.043***
	(0.010)	(0.002)
Number of servants	1.018***	0.150***
	(0.003)	(0.000)
Constant	-4.132***	
	(0.010)	
Observations	7,213,217	
Pseudo R2	0.193	
Chi-squared	887,072.910	
p-value	0.000	

*** $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$**

Weights ($\boldsymbol{w}$, second column) and marginal effects ($\boldsymbol{\partial}\mathbf{y}/\boldsymbol{\partial}\mathbf{x}$, third column) calculated from the 1891 training data for same features and labels of the previous tables but without SubOccode for computing efficiency. Note that the Female weight is now positive. In parentheses and below, standard errors (**se**).

Table 5: MNL model weights without SubOccode.

	Labels: W/E/OA			
Feature	**Weight ($\boldsymbol{w}$)**	***t*-stat**	**Weight ($\boldsymbol{w}$)**	***t*-stat**
	2.Employer		**3.Own account**	
RSD Density	-0.00745***	(-105.62)	-0.00365***	(-72.43)
RSD Density × RSD Density	0.0000217***	(80.07)	0.00000929***	(42.80)
Age	0.119***	(158.85)	0.114***	(229.08)
Age × Age	-0.000793***	(-101.43)	-0.000778***	(-145.02)
Sex				
1. Male	0	(.)	0	(.)
2. Female	-0.271***	(-26.55)	0.947***	(206.47)
Marital status				
1. Single	0	(.)	0	(.)
2. Married	0.0396***	(4.74)	-0.504***	(-89.45)
4. Widowed	-0.283***	(-26.28)	-0.290***	(-38.29)
2. Female × 2. Married	0.115***	(7.44)	0.370***	(48.25)
2. Female × 4. Widowed	0.439***	(30.54)	0.00476	(0.57)
Relationship to the head				
1. Head	0	(.)	0	(.)
2. CFU member	-1.029***	(-108.67)	-0.872***	(-161.54)
3. Older generation	-1.242***	(-50.43)	-1.087***	(-73.87)
4. Siblings	-0.845***	(-52.84)	-0.785***	(-89.18)
5. Other family	-1.464***	(-46.07)	-0.962***	(-75.13)
6. Servants	-3.173***	(-29.52)	-3.268***	(-67.07)
7. Working title	-2.162***	(-25.26)	-2.276***	(-60.20)
8. Lodgers/boarders	-1.480***	(-119.60)	-1.158***	(-171.50)
9. Non-household	-1.829***	(-35.02)	-1.493***	(-54.88)
10. Unknown	-0.171***	(-9.08)	-0.257***	(-22.05)
Number of servants	1.430***	(361.84)	0.705***	(212.89)
Constant	-5.535***	(-308.16)	-4.390***	(-384.13)
Observations		7,173,550		

t statistics in parentheses

*** $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$**

Weights ($\boldsymbol{w}$, second and fourth columns) calculated from the 1891 training data for the features without SubOccode for ease of computation.is now positive. Multi-class labels are Worker (W), Employer (E), and Own account (OA) (Worker is base category). In parentheses (third and fifth columns), ***t*-stat**).

Table 6: MNL model marginal effects without the SubOccode.

	Labels: W/E/OA		
Feature	$\partial y/\partial x$/**se**	$\partial y/\partial x$/**se**	$\partial y/\partial x$/**se**
RSD Density	0.001***	-0.000***	-0.000***
	(0.000)	(0.000)	(0.000)
Age	-0.009***	0.002***	0.007***
	(0.000)	(0.000)	(0.000)
Sex			
1. Male	0.000	0.000	0.000
	(0.000)	(0.000)	(0.000)
2. Female	-0.143***	-0.014***	0.157***
	(0.001)	(0.000)	(0.001)
Martial status			
1. Single	0.000	0.000	0.000
	(0.000)	(0.000)	(0.000)
2. Married	0.042***	0.005***	-0.047***
	(0.001)	(0.000)	(0.001)
4. Widowed	0.038***	-0.004***	-0.034***
	(0.001)	(0.000)	(0.001)
Relationship to the head			
1. Head	0.000	0.000	0.000
	(0.000)	(0.000)	(0.000)
2. CFU member	0.136***	-0.038***	-0.098***
	(0.001)	(0.000)	(0.001)
3. Older generation	0.157***	-0.043***	-0.114***
	(0.001)	(0.001)	(0.001)
4. Siblings	0.124***	-0.033***	-0.090***
	(0.001)	(0.001)	(0.001)
5. Other family	0.152***	-0.048***	-0.104***
	(0.001)	(0.001)	(0.001)
6. Servants	0.246***	-0.063***	-0.183***
	(0.001)	(0.000)	(0.001)
7. Working title	0.224***	-0.057***	-0.167***
	(0.001)	(0.001)	(0.001)
8. Lodgers/boarders	0.167***	-0.048***	-0.119***
	(0.001)	(0.000)	(0.001)
9. Non-household	0.192***	-0.053***	-0.139***
	(0.002)	(0.001)	(0.001)
10. Unknown	0.042***	-0.007***	-0.035***
	(0.002)	(0.001)	(0.002)
Number of servants	-0.126***	0.053***	0.073***
	(0.000)	(0.000)	(0.000)
Constant		-4.132***	
		(0.010)	
Observations		7,173,550	
Pseudo R2		0.185	
Chi-squared		995,776.903	
p-value		0.000	

*** $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$**

Marginal effects ($\partial \mathbf{y}/\partial \mathbf{x}$) calculated from the 1891 training data for same features and labels of the previous tables but without SubOccode for computing efficiency. Multi-class labels are Worker (W), Employer (E), and Own account (OA) In parentheses and below, standard errors (**se**).

Appendix A. Mathematical Appendix

As first established by Goldberger (1991), the function of interest is the *conditional expectation function* (CEF) which in the case of a given value $\boldsymbol{i}$ of a binary label, $\boldsymbol{y_i}$, is the probability that the label is 1 given the value of a feature $\boldsymbol{i}$, $\boldsymbol{x_i}$ as presented in Rabe-Hesketh and Skrondal (2012):

$$\boldsymbol{E(y_i|\mathrm{x}) = \mathrm{Pr}(y_i = 1|\mathrm{x})}$$

The probability must lie between 0 and 1, thus a non-linear link function is used to estimate the following linear relation (Rabe-Hesketh and Skrondal, 2012):

$$\mathrm{link}\boldsymbol{\{\mathrm{Pr}(y_i = 1|\mathrm{x})\} = bias + weights'\ \mathrm{x}}$$

where the intercept is called *bias* and the slope coefficients are called *weights* (Murphy, 2012). Sometimes bias and weights together are also called *parameters* (Goodfellow et al., 2016). The link function that we use in this paper is the *logit* defined as the logarithm of the ***odds*** by Rabe-Hesketh and Skrondal (2012):

$$\mathrm{logit}\boldsymbol{\{\mathrm{Pr}(y_i = 1|\mathrm{x})\}} \equiv \mathrm{logarithm}\,\boldsymbol{\{odds(y_i = 1|\mathrm{x})\} = bias + weights'\ \mathrm{x}}$$

and the odds that the label is one are defined as follows (Rabe-Hesketh and Skrondal, 2012):

$$\boldsymbol{odds(y_i = 1|\mathrm{x}) \equiv \frac{\mathrm{Pr}(y_i = 1|\mathrm{x})}{1 - \mathrm{Pr}(y_i = 1|\mathrm{x})}}$$

Taking the inverse of the logit function makes possible to estimate the probability that the label is one given a certain value of the feature (Rabe-Hesketh and Skrondal, 2012):

$$\mathrm{Pr}\boldsymbol{(y_i = 1|\mathrm{x}) = \mathrm{logit}^{-1}(bias + weights'\ \mathrm{x}) \equiv \frac{\mathrm{exp}^{bias+weights'\ \mathrm{x}}}{1 + \mathrm{exp}^{bias+weights'\ \mathrm{x}}}}$$

Being $\Pr(\boldsymbol{y_i} = \mathbf{0}|\mathrm{x}) = \mathbf{1} - \Pr(\boldsymbol{y_i} = \mathbf{1}|\mathrm{x})$, then also:

$$\Pr(\boldsymbol{y_i} = \mathbf{0}|\mathrm{x}) = \frac{\mathbf{1}}{\mathbf{1} + \mathbf{exp}^{\boldsymbol{bias}+\boldsymbol{weights'}\,\mathrm{x}}}$$

The logit model forms part of the so-called *single-index models* where the CEF is equal to a non-linear mean function $\boldsymbol{F}()$ (i.e., the inverse of the logit, or logit$^{-\mathbf{1}}$) of a single index, $\boldsymbol{weights'}\ \mathrm{x}$, of the features and the weights, following Cameron and Trivedi (2005):

$$\boldsymbol{E}(\boldsymbol{y}|\mathrm{x}) = \boldsymbol{F}(\boldsymbol{weights'}\ \mathrm{x})$$

And the effect on the CEF of a change in the ith regressor is, according to Cameron and Trivedi (2005):

$$\frac{\boldsymbol{\partial E}(\boldsymbol{y}|\boldsymbol{x_i})}{\boldsymbol{\partial x_i}} = \boldsymbol{F'}(\boldsymbol{weights'}\ \mathrm{x})\boldsymbol{weight_i}$$

and $\boldsymbol{F'} = \frac{\boldsymbol{\partial F}()}{\boldsymbol{\partial}}$. Thus the *relative effect* of changes in regressors is equal to the ratio of the weights, also following Cameron and Trivedi (2005):

$$\frac{\boldsymbol{\partial E}(\boldsymbol{y}|\boldsymbol{x_i})/\boldsymbol{\partial x_i}}{\boldsymbol{\partial E}(\boldsymbol{y}|\boldsymbol{x_k})/\boldsymbol{\partial x_k}} = \frac{\boldsymbol{weight_i}}{\boldsymbol{weight_k}}$$

as $\boldsymbol{F'}(\boldsymbol{weights'}\ \mathrm{x})$ cancels in the numerator and the denominator. This means that, for instance, if $\boldsymbol{weight_i}$ is three times $\boldsymbol{weight_k}$, then a one unit change of $\boldsymbol{x_i}$ has three times the effect of a one-unit change in $\boldsymbol{x_k}$. And if $\boldsymbol{F}()$ is monotonic, as the inverse logit is, then the signs of the weights command the signs of the effects for all possible values of the feature.

In the case of a multi-class label, for example, if we are classifying individuals into entrepreneurial status of E, OA or non-entrepreneurs as W, a general model can be written following Cameron and

Trivedi (2005) where $\boldsymbol{J}$ is a given category out of the $\boldsymbol{j}$ categories and $\boldsymbol{j}$ goes from $\mathbf{1}$ to $\boldsymbol{C}$:

$$\Pr(\boldsymbol{y_j} = \boldsymbol{J}|\mathrm{x}) = \frac{\mathbf{exp}^{\boldsymbol{bias}^{[J]}+\boldsymbol{weights}^{[J]}\ \mathrm{x}}}{\sum_{j=1}^{C} \mathbf{exp}^{\boldsymbol{bias}^{[j]}+\boldsymbol{weights}^{[j]}\ \mathrm{x}}}$$

where the superscript enclosed in the square brackets, [], is used to signal that the weights and the bias pertain to a given class, $\boldsymbol{j}$ and the denominator is equal to the sum of the numerators, so that the probabilities sum up to one. Consequently, this is a MNL with alternative-invariant features with alternative-specific weights. Also, the first, usually the most frequent category, is taken as base category which means the following two important assumptions: $\boldsymbol{bias}^{[\mathbf{1}]} = \mathbf{0}$ and $\boldsymbol{weight}_{\boldsymbol{i}}^{[\mathbf{1}]} = \mathbf{0}$, so each weight must be interpreted as the change in probability that a unit increase in a given feature *with respect to* the base category. For the case, where $\boldsymbol{C} = \mathbf{3}$ with categories $(\boldsymbol{j} = \mathbf{1}) \equiv \boldsymbol{Worker}$, $(\boldsymbol{j} = \mathbf{2}) \equiv \boldsymbol{Employer}$, and $(\boldsymbol{j} = \mathbf{3}) \equiv \boldsymbol{Own\ account}$ the probabilities are as follows using Rabe-Hesketh and Skrondal (2012):

$$\Pr(\boldsymbol{y_j} = \mathbf{1}|\mathrm{x}) = \frac{\mathbf{1}}{\mathbf{1} + \mathbf{exp}^{\boldsymbol{bias}^{[2]}+\boldsymbol{weights}^{[2]}\ \mathrm{x}} + \mathbf{exp}^{\boldsymbol{bias}^{[3]}+\boldsymbol{weights}^{[3]}\ \mathrm{x}}}$$

$$\Pr(\boldsymbol{y_j} = \mathbf{2}|\mathrm{x}) = \frac{\mathbf{exp}^{\boldsymbol{bias}^{[2]}+\boldsymbol{weights}^{[2]}\ \mathrm{x}}}{\mathbf{1} + \mathbf{exp}^{\boldsymbol{bias}^{[2]}+\boldsymbol{weights}^{[2]}\ \mathrm{x}} + \mathbf{exp}^{\boldsymbol{bias}^{[3]}+\boldsymbol{weights}^{[3]}\ \mathrm{x}}}$$

$$\Pr(\boldsymbol{y_j} = \mathbf{3}|\mathrm{x}) = \frac{\mathbf{exp}^{\boldsymbol{bias}^{[3]}+\boldsymbol{weights}^{[3]}\ \mathrm{x}}}{\mathbf{1} + \mathbf{exp}^{\boldsymbol{bias}^{[2]}+\boldsymbol{weights}^{[2]}\ \mathrm{x}} + \mathbf{exp}^{\boldsymbol{bias}^{[3]}+\boldsymbol{weights}^{[3]}\ \mathrm{x}}}$$

This is a discrete choice model to predict the employment status of any economically active individual in the census using, as previously said, alternative-invariant features with alternative-specific weights.